VBA
voor
ACCESS
2003

19 juni 2009
Werfkade 10
9601LG Hoogezand
Tel. 0598 390070
e-mail: bti@bbti.nl

Inhoudsopgave

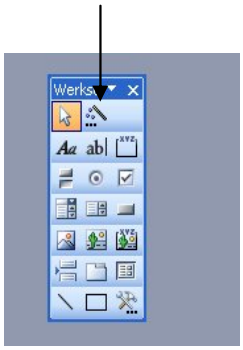
Eerste ronde

Sessie 1 - 1

VBA gebruiken.

Met Visual Basic kun je de mogelijkheden van Access enorm uitbreiden. We gaan in iedere sessie ook even specifiek kijken naar de VBA-taal en -mogelijkheden.

- Zet een nieuwe database op, noem hem maar **OefenDB** en bewaar hem in een eigen map (bijv. **accessvba**).
- Start een nieuw formulier in ontwerpweergave, nergens op gebaseerd. Maak het wat groter, 10 bij 8 bijvoorbeeld. Bewaar het even als **frmTest**.



We gaan nu wat formulierobjecten plaatsen met behulp van de wel ergens aanwezige werkset.

- Zorg ervoor dat de Wizard in de werkset uitgeschakeld is.
- Selecteer de **Opdracht**knop en plaats een Opdracht-knop rechts onder in je formulier (**Annuleer** wanneer je navigatiekeuzen moet maken):

We kunnen die knop nu zelf een functie geven:

- Selecteer de knop en dan rechter muisknop en kies **Eigenschappen**.
- Zorg er hiervoor dat **Alle** zichtbaar zijn.

Je krijgt bovenaan:

De Naam is de echte naam van het object. Wil je het object manipuleren dan gebruik je deze naam. Het bijschrift is wat er op de knop staat.

- ☼ Geef hem de **Naam**: cmdTelOp en het bijschrift Tel op!

We gebruiken **cmd** hier om aan te geven dat het om een command-knop gaat. Omdat we nu gaan programmeren is **cmd** wellicht te prefereren boven **btn** (voor button). Het maakt geen wezenlijk verschil. Gebruik kleine letters. Dit is niet meer dan programmeurs traditie en helpt je om later gauw te zien wat wat is.

- ☼ Sluit en noem hem **frmOptellen. K**ijk in weergavemodus naar je formulier.

De knop doet nog weinig.

Visual Basic weet nog niet dat het formulier bestaat, kijk maar.

- ☼ Ga naar de VBA-editor (**Extra/Macro/Visual Basic Editor**) en merk op dat het formulier niet voorkomt waar het onder **OefenDB** had moeten staan.

VBA-weet het pas zodra daar aanleiding toe is.

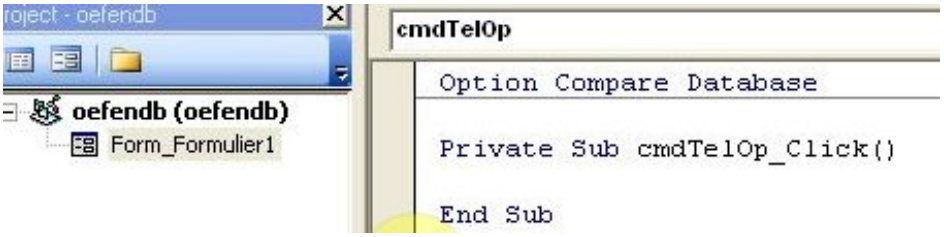
- ☼ Ga weer naar de **Eigenschappen** van de cmdTelOp-knop. (Je hoeft hiervoor het vba-venster niet te sluiten!)

- ☼ Selecteer het tabblad **Gebeurtenis** en selecteer uit de lijst: **Bij klikken**. Klik daarna op de drie puntjes (…)

Je krijgt:

- ☼ Kies **Opbouwfunctie voor programmacode**

En voilà, je zit meteen in de Editor en je formulier is toegevoegd. De basis VBA-code zie je rechts.



Op het moment dat iemand op die knop klikt, kunnen wij de zaak dus overnemen. Maar eens proberen.

- ☼ Schakel terug naar het formulier (je hoeft de Editor niet te sluiten!) en voeg een Labelveld (uit de Werkset) toe. Type er direct ook iets in, bijv. "Mijn tekst".

Nu kunnen we met een druk op de opdrachtknop iedere gewenste tekst in die label laten verschijnen. Maar eerst even de Labelknop benoemen.

- ☼ Geef de knop de Naam: lblTelOP.

Het leven na een foutje.

- ☼ Nu naar de opdrachtknopprocedure (Gewoon via de Microsoft Visual Basic Editor knop onder in je scherm) :

- ☼ Pas aan zodat er staat:

Private Sub cmdTelOp_Click()
lblTelOp.Bijschrift = "Uitkomst van de optelling"
End Sub

We hadden hier Caption moeten gebruiken i.p.v. Bijschrift, maar doen dat nu even met opzet om te zien wat je moet doen als het niet goed gaat.

- ☼ *Ga terug naar je formulier en probeer*

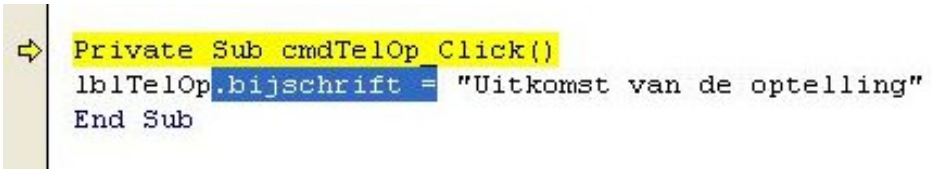
Je krijgt de volgende foutmelding:

```
Private Sub cmdTelOp_Click()
lblTelOp.bijschrift = "Uitkomst van de optelling"
End Sub
```

Aan de tekst met de blauwe achtergrond, mankeert dus iets. Wat dat is weten we ondertussen wel. Het had *Caption* moeten zijn. Caption is de Engelse term is voor Bijschrift. Wat hier dus staat is: zodra je op die knop klikt moet de tekst "Dag klein wereldtje" in de label verschijnen.

- ☼ Klik op *OK* om verder te kunnen.

Je krijgt nu de code met een gele regel:



Dit is een belangrijk moment, want als er zich een fout heeft voorgedaan, blijft het programma als het ware ergens haken en kunnen er van allerlei onverwachte dingen gebeuren als je toch zomaar verder gaat. Na iedere fout moet het programma "losgepeuterd" worden. En dat doe je bovenaan in de menubalk:

- ☼ Klik op het blauwe vierkante knopje ("beginwaarden").

Dan kun je nu je code aanpassen en kunnen we ongestraft verder gaan.

- ☼ Vervang **Bijschrift** door **Caption**.

De tekst **lblTelOp.Caption = "Dag …"** zegt dus eigenlijk: het object **lblTelOP** heeft een eigenschap die heet **Caption** en die eigenschap kan gevuld worden met een tekst. Je scheidt de objectnaam en de eigenschap met een punt (.).

- ☼ Probeer.

- ☼ Probeer ook even met een getal (getallen niet tussen aanhalingstekens plaatsen!), bijv.:

lblTelOp.Caption = 99

We willen VBA even laten optellen. Het antwoord kan dan mooi in het labelveld verschijnen.

Sessie 1 - 2

Rekenen

- ☼ Plaats twee Tekstvakken, geef ze de *naam* **txtGetal1** en **txtGetal2**. Vul de bijbehorende labeltjes links van de tekstvelden met "**Getal 1**" en "**Getal 2**". Van de tekst in de label bovenaan heb ik ook maar "**Uitkomst**" gemaakt en de naam van de label heb ik aangepast en hem **lblTelOp** gedoopt.

Zoiets:

- ☼ Pas de code achter de cmdTelOp-knop nu als volgt aan:

```
Private Sub cmdTelOp_Click()
getal1 = txtGetal1.Text
getal2 = txtGetal2.Text
som = getal1 + getal2
lblTelOp.Caption = som
End Sub
```

We maken hierbij gebruik van **Variabelen**. Variabelen zijn benoemde plaatsen in het geheugen waar je iets kunt bewaren, kleine klembordjes dus. De opdracht getal1=txtGetal1.Text zegt dus: bewaar het getal dat er in dat tekstvak (de eigenschap Text van het object TxtGetal1) staat even in een geheugenplekje dat ik **getal1** noem. De som van die twee getallen stoppen we dan in de label. We gaan dit zo even proberen en zullen dan merken dat het nog niet helemaal werkt. Let zo even op de foutmelding.

- ☼ Probeer (type 2 getallen in en klik op de opdrachtknop).

De foutmelding luidt:

Hoewel dit niet geldt voor het Labelobject wil het tekstvak in ieder geval eerst *de focus* hebben voordat het z'n geheimen prijs geeft. Hoe zou je die focus kunnen plaatsen? Je zou denken dat daar wel een methode voor is.

- ☼ **[Beeïndig]** en peuter los.

- ☼ Type op de regel boven Getal1 = etc: txtGetal1.

Zo:

```
Private Sub cmdTelOp_Click()
txtGetal1.
getal1 = txtGetal1.Text
```

Na het intypen van de punt krijg je allemaal mogelijkheden. Je kunt blijven typen, maar de gewenste uitdrukking ook uit het lijstje halen.

- ☼ Selecteer *Setfocus*

- ☼ Doe hetzelfde voor het toekennen van Getal2.

Je procedure ziet er nu als volgt uit:

```
Private Sub cmdTelOp_Click()
txtGetal1.SetFocus
getal1 = txtGetal1.Text
txtGetal2.SetFocus
getal2 = txtGetal2.Text
lblTelOp.Caption = getal1 + getal2
End Sub
```

Er blijft nog een probleempje over zoals we zo zullen zien.

- ☼ Probeer je programma en let even op de uitkomst.

Dit is een aparte manier van optellen waar een boekhouder niet gelukkig mee zal zijn. De getallen worden als tekst beschouwd. We kunnen dit o.a. oplossen door van de variabelen specifieke **getalvariabelen** te maken. Je moet dat van te voren aan vba kenbaar maken en dat doe je met de opdracht *Dim getal1 as Integer*. Daarna verwacht VBA dat er in *getal1* een heel getal zit. Dit even melden, heet in het jargon ***Declareren***

- ☼ Pas de code aan zodat je het volgende krijgt:

```
Private Sub cmdTelOp_Click()
Dim getal1 As Integer
Dim getal2 As Integer
Dim som as Integer
txtGetal1.SetFocus
getal1 = txtGetal1.Text
txtGetal2.SetFocus
getal2 = txtGetal2.Text
lblTelOp.Caption = getal1 + getal2
End Sub
```

- ☼ Probeer.

Methoden en eigenschappen

- ☼ Verwijder .SetFocus nogmaals uit txtGetal1.SetFocus en type de punt weer in.

- ☼ Druk nu op de S en je zit al direct een stuk dichter bij de SetFocus methode. Lees even door.

SetFocus wordt voorafgegaan door de groene icoon, dit markeert een Methode. De andere icoontjes (met het handje) markeren Eigenschappen (Properties) van een object. Objecten hebben dus Eigenschappen en Methoden. Typisch voor de eigenschap van een object is dat je er "=" achter kunt zetten, bijv. txtGetal1.text = "mijn tekst". Bij txtGetal.Setfocus kan dat niet. Je kunt van een fiets zeggen: "De fiets = Zwart" Je hebt het dan over een **eigenschap**

van de fiets. Wil je zeggen: "De fiets rijdt" dan is "rijdt" in programmeertermen een **methode** van de fiets. Methoden zijn dus het meest actie-gericht, hoewel je niet moet vergeten dat dit een kunstmatig onderscheid is dat door de programmataal-ontwikkelaars gemaakt is. In plaats van Object.Setfocus hadden ze ook kunnen kiezen voor Object.Focus = True waarmee de focusbepaling een eigenschap was geworden. Het blijft dus lichtjes arbitrair, maar we hebben toch geen andere keus dan de methoden en eigenschappen gebruiken zoals ze ingesteld zijn.

- ☼ Plaats **SetFocus** weer terug en sla alles even op.

Declareren en Dimensioneren

Het aangeven hoe iets heet, noemen ze DECLAREREN in het ontwikkeljargon. *Private Sub cmdTelOp_Click()* vertelt VBA dat er een procedure is die "cmdTelOp_Click" heet. Ook als je variabelen gebruikt moet de naam ervan bekend gemaakt worden. Nu doet VBA dat wel voor je als je het vergeet, maar dat heeft ook beperkingen. Zodra jij zegt *Getal1 = 9* registreert VBA gelijk dat er dus een variabele bestaat die "Getal1" heet. Door dat bekendmaken eerst even zelf netjes te regelen heb je wat meer controle, je kunt bijvoorbeeld al direct aangeven dat je zo'n variabele voor een getal wilt gebruiken en niet voor tekst. Dat scheelt want dan hoeft er minder ruimte gereserveerd te worden. VBA reserveert relatief veel ruimte voor een variabele als hij niet weet waar ie voor gebruikt gaat worden. Zo'n variabele heet een **Variante** variabele. Je neemt het zelf ter hand door zo'n variabele te DIMENSIONEREN. Dit gebeurt met het Dim-statement, bijv. Dim Getal1 as Integer geeft aan dat je de variabele gaat gebruiken voor hele getallen (tussen -32.768 en +32.767). Zo hadden we dus het probleem met de getallen in de laatste oefening ook op kunnen lossen. Met andere woorden je gebruikt een Dim-statement om een variabele te declareren. Zo'n Dim-statement moet natuurlijk verschijnen voordat hij gebruikt wordt.

Hier is een lijstje van Variabele-typen:

Gegevenstype	Lengte	Bereik
Byte	1 byte	0 tot 255
Boolean	2 bytes	**True** of **False**
Geheel getal	2 bytes	-32.768 tot 32.767
Long (lange integer)	4 bytes	-2.147.483.648 tot 2.147.483.647
Single (enkelvoudige precisie met drijvende komma)	4 bytes	-3,402823E38 tot -1,401298E-45 voor negatieve waarden; 1,401298E-45 tot 3,402823E38 voor positieve waarden
Double (dubbele precisie met drijvende komma)	8 bytes	-1,79769313486232E308 tot en met -4,94065645841247E-324 voor negatieve waarden; 4,94065645841247E-324 tot 1,79769313486232E308 voor positieve waarden.
Currency (geschaalde integer)	8 bytes	-922.337.203.685.477,5808 tot 922.337.203.685.477,5807
Decimal	14 bytes	+/-79.228.162.514.264.337.593.543.950.335 zonder decimale komma; +/-7,9228162514264337593543950335 met 28 plaatsen rechts van de decimale komma; het kleinste getal met de waarde niet gelijk aan nul is:
Date	8 bytes	1 januari 100 tot 31 december 9999
Object	4 bytes	Elke verwijzing naar **Object**
String (variabele lengte)	10 bytes + lengte van tekenreeks	0 tot ongeveer 2 miljard
String (vaste lengte)	Lengte van tekenreeks	1 tot ongeveer 65.400
Variant (met getallen)	16 bytes	Elke numerieke waarde binnen het bereik van het gegevenstype **Double**
Variant (met tekens)	22 bytes + lengte van tekenreeks	Hetzelfde bereik als voor **String** met variabele lengte
Door de gebruiker gedefinieerd (met **Type**)	Getal dat is vereist door de elementen	Het bereik van elk element is hetzelfde als het bereik van het betreffende gegevenstype.

Merk op dat je een variabele ook kunt gebruiken om naar een Object te verwijzen. Die variabele bevat dan (= onthoudt dan) over welk object het gaat. We zullen straks zien hoe handig dat is bij het werken met Access-objecten.

Sessie 1 - 3

Dan gaan we nu eens zien hoe wie deze kennis kunnen gebruiken om met data te werken.

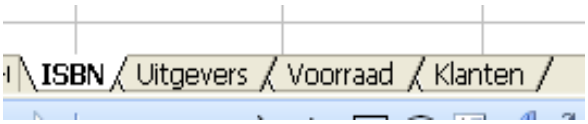
We hebben een postorderbedrijf dat boeken verkoopt. Het bedrijf heet BOEKENIER.
Tot nu toe stonden onze gegevens in een Excel werkmap. We willen die overhevelen naar Access.

Importeren uit Excel

☼ Download het bestand **boekenier**.xls van de cursus website.

☼ Open het bestand **boekenier.xls** in Excel.

Het bestaat uit vier bladen:



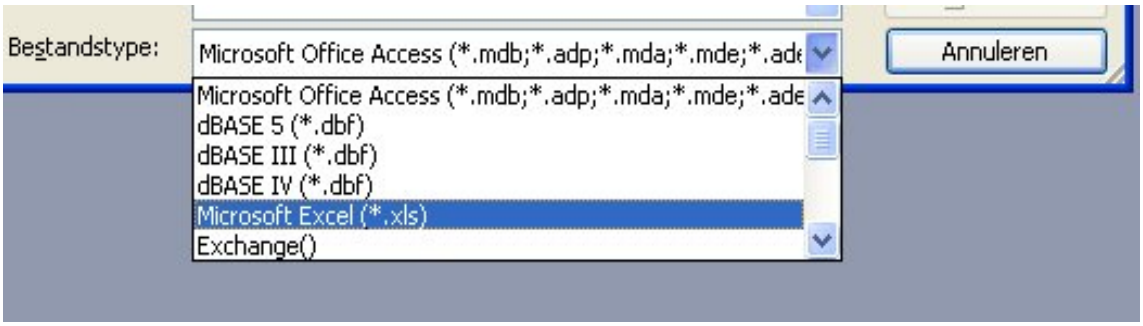
☼ Bekijk de gegevens.

We gaan deze data nu overbrengen naar vier verschillende tabellen in Access.

☼ Start **Access** en een nieuwe database. Bewaar die als **boekenier.mdb** op je stick of verkozen cursusmap.

☼ Kies nu **Bestand/Externe gegevens ophalen/Importeren**

☼ Verander het bestandstype in Excel (*. xls) en ga naar de map met het boekenier-bestand.



☼ Kies **boekenier.xls** en klik op **Importeren**

Je krijgt nu de vier Excelbladen te zien:

- ☼ Selecteer **ISBN** en klik op **Volgende**
- ☼ Laat weten dat de eerste rij de koppen bevat en weer **Volgende**
- ☼ We kiezen voor opslaan in een nieuwe tabel en dus alleen maar **Volgende**.
- ☼ Het volgende scherm accepteren we als standaard en weer **Volgende**
- ☼ We kunnen het ISBN-veld wel als sleutel gebruiken.
- ☼ Pas dat dus aan en **Volgende**

Laten we de tabel waarin de gegevens geïmporteerd worden maar **tblISBN** noemen.

- ☼ Pas dat aan en voltooien
- ☼ Importeer op dezelfde manier de gegevens uit het Excelblad *Uitgevers* in een nieuwe Access tabel met de naam **tblUitgevers** en *Klanten* in **tblKlanten**. Laat Access voor deze records zelf maar een sleutel maken.
- ☼ Importeer *Voorraad* in **tblVoorraad** met ISBN als sleutel.

Waar nodig gaan we deze gegevens nog wel aanpassen.

Sessie 1 - 4

- ☼ Start een tweede formulier en bewaar het als **frmUitgevers.**

We gaan nu een knop gebruiken om gegevens uit een tabel te halen.

☼ Plaats een knop (**cmdUitgevers / Caption: "Toon uitgevers!"**) en een label (**lblUitgevers**) . Plaats een paar tijdelijke streepjes in de label.

☼ Zorg voor de basis opdrachtcode:

```
Private Sub cmdUitgevers_Click()

End Sub
```

☼ Vul deze nu met de volgende code (in de volgende ronde kom ik terug op een aantal aspecten van deze code):

```
Private Sub cmdUitgevers_Click()
Dim db As Database
Dim rec As Recordset
Dim AantalRecords As Integer
Set db = CurrentDb()
Set rec = db.OpenRecordset("tblUitgevers")
AantalRecords = rec.RecordCount
lblUitgevers.Caption = AantalRecords
rec.Close
End Sub
```

☼ Probeer. (Het resultaat zou 4 moeten zijn, want we hebben 4 uitgevers in ons tabelletje.)

Sessie 1 – 5

En nu eens zien of we op deze manier ook gegevens uit die tabel kunnen halen

Pas de code als volgt aan:

```
Private Sub cmdUitgevers_Click()
Dim db As Database
Dim rec As Recordset
Dim aantalRecords As Integer
Dim uitgever as String
Set db = CurrentDb()
Set rec = db.OpenRecordset("tblUitgevers")
aantalRecords = rec.RecordCount
rec.MoveFirst
uitgever = rec.Fields(1)
MsgBox (uitgever)
lblUitgevers.Caption = aantalRecords
rec.Close
End Sub
```

De methode *rec.movefirst* verplaatst de focus (de "cursor" in database terminologie) naar de eerste record van de tabel.

De eigenschap *rec.fields(1)* bevat de inhoud van het tweede (!) veld (het veld "uitgever"). In de ICT hebben ze nu eenmaal de gewoonte om bij 0 te beginnen. Met de opdracht *uitgever = rec.fields(1)* plaats je de uitgever (uit het eerste record) in de variabele *uitgever*.

De VBA-opdracht *msgbox(uitgever)* toont een klein venstertje met de inhoud van wat er in die variabele zit. Let er op als je *msgbox("uitgever")* schrijft (met aanhalingstekens dus) dan krijg je straks de boodschap ***uitgever*** te zien, de tekst zelf dus en niet de inhoud van de variabele.

☼ Probeer!

Je kunt de cursor verplaatsen naar de tweede record, met de methode *rec.movenext*

☼ Pas als volgt aan en probeer.

```
Private Sub cmdUitgever_Click()
Dim db As Database
Dim rec As Recordset
Dim aantalRecords As Integer
Set db = CurrentDb()
Set rec = db.OpenRecordset("tblUitgevers")
aantalRecords = rec.RecordCount
rec.MoveFirst
MsgBox (rec.Fields(1))
rec.MoveNext
MsgBox (rec.Fields(1))
rec.MoveNext
MsgBox (rec.Fields(1))
rec.MoveNext
MsgBox (rec.Fields(1))
lblUitgevers.Caption = aantalRecords
rec.Close
End Sub
```

Extra opdracht:

☼ Hoe zou je ervoor kunnen zorgen dat de MsgBox alle NAW gegevens van het eerste record de een na de ander toont?

☼ Is RecordCount een methode of een eigenschap? Waarom? En MoveFirst?

Met MoveLast verplaats je de cursor naar de laatste record. Het is aan te raden om deze opdracht altijd toe te voegen, zeker bij grote en gekoppelde tabellen. Dit komt omdat Access soms de records niet allemaal "kent" totdat het einde van de tabel bekend is. Je krijgt dan bijvoorbeeld een foutief aantal records.

☼ Voeg deze methode toe, maar op zo'n manier dat je code toch bij de eerste record begint om de gegevens te tonen.

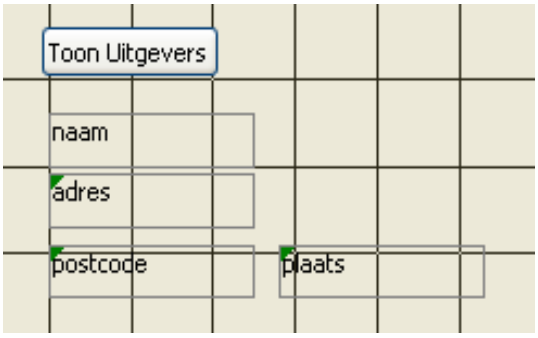
Tweede Ronde

Sessie 2 - 1

Navigeren door tabel-records.

Om de volledige adressen in één keer te tonen, zou je in de eerste plaats vier labelvelden nodig hebben. In plaats van in de messagebox,zet je die gegevens dan in die labels.

☼ Creëer de velden, zoiets (**lblNaam/lblAdres/lblPc/lblPlaats**):



Je code gaat dan als volgt worden:

```
Private Sub cmdUitgever_Click()
Dim db As Database
Dim rec As Recordset
Dim aantalRecords As Integer
Set db = CurrentDb()
Set rec = db.OpenRecordset("tblUitgevers")
rec.MoveLast
aantalRecords = rec.RecordCount
rec.MoveFirst
info = rec.Fields(1)
lblNaam.Caption = info
info = rec.Fields(2)
lblAdres.Caption = info
info = rec.Fields(3)
lblPc.Caption = info
info = rec.Fields(4)
lblPlaats.Caption = info
lblUitgevers.Caption = aantalRecords
rec.Close
End Sub
```

☼ Pas je code aan.

Het Dim-statement wordt hier gebruikt om de variabelen db, rec, en aantalrecords te declareren (= bij Access bekend maken wat voor soort variabele het is). Je mag ook best

andere namen gebruiken, maar "db" is gebruikelijk om met een variabele naar een database te verwijzen, en "rec" naar de verzameling records van een tabel. Met Set db = en Set rec = … wordt dan aangegeven om welke database en welke tabel het specifiek gaat. Het grote voordeel is dat je daarna met alleen db naar een database kunt verwijzen en met alleen maar rec naar de records van een bepaalde tabel.
Set db = CurrentDb() geeft aan dat we met de huidige (dus geen andere, externe) database willen werken. *Set rec = db.OpenRecordset("tblUitgevers")* geeft aan dat we met de tabel "tblUitgevers" willen werken. Van nu af aan, kunnen we bijv. rec.Fields(1) gebruiken en weet de software dat we het eerste veld van een record van tblUitgevers willen hebben.

☼ Voer de code uit.

Om de rest van de gegevens te tonen, kun je naar de volgende record verhuizen en dan weer hetzelfde verhaal gebruiken. Je gaat naar de volgende record met de .MoveNext methode. Dat zou dus zoiets moeten worden: (Je kunt er een messageboxje tussen laten om ze een voor een te zien)

```
Private Sub cmdUitgever_Click()
    Dim db As Database
    Dim rec As Recordset
    Dim aantalRecords As Integer
    Dim info as string
    Set db = CurrentDb()
    Set rec = db.OpenRecordset("tblUitgevers")
    rec.MoveLast
    aantalRecords = rec.RecordCount
    rec.MoveFirst
    info = rec.Fields(1)
    lblNaam.Caption = info
    info = rec.Fields(2)
    lblAdres.Caption = info
    info = rec.Fields(3)
    lblPc.Caption = info
    info = rec.Fields(4)
    lblPlaats.Caption = info
    Msgbox("Volgende")
    rec.MoveNext
    info = rec.Fields(1)
    lblNaam.Caption = info
    info = rec.Fields(2)
    lblAdres.Caption = info
    info = rec.Fields(3)
    lblPc.Caption = info
    info = rec.Fields(4)
    lblPlaats.Caption = info
    lblUitgevers.Caption = aantalRecords
    rec.Close
End Sub
```

☼ Gebruik kopiëren en plakken om de code zoals hierboven aan te passen. Probeer.

Dit wordt helemaal leuk als je een paar duizend records hebt. Voordat je het weet ben je dan een echte kopieer en plak expert! (Of je hebt RSI) Maar het kan ook anders, we houden ons immers bezig met automatiseren!

Lussen (for .. next)

Als je dezelfde, of bijna dezelfde code meerdere keren nodig hebt, kun je gebruik maken van de lus-structuur. Die ziet er bijvoorbeeld zo uit:

```
For A = 1 to 10
Doe iets.
Next A
```

Met deze opdrachtstructuur, wordt er 10 keer iets gedaan. In plaats van 1 en 10 kun je allerlei gewenste getallen invullen. A = hier een variabele (met een korte naam, je mag ook wel een langere naam gebruiken). Hierin wordt bijgehouden hoe vaak een opdracht al uitgevoerd is. Bij de eerste keer zit er 1 in A, bij de tweede 2 etc. tot het maximum bereikt is, dan loop het programma weer verder. Zo kunnen we onze code dus mooi vereenvoudigen en toch alle adressen tonen:

```
Private Sub cmdUitgever_Click()
Dim db As Database
Dim rec As Recordset
Dim aantalRecords As Integer
Set db = CurrentDb()
Set rec = db.OpenRecordset("tblUitgevers")
rec.MoveLast
aantalRecords = rec.RecordCount
rec.MoveFirst
For A = 1 To aantalRecords
info = rec.Fields(1)
lblNaam.Caption = info
info = rec.Fields(2)
lblAdres.Caption = info
info = rec.Fields(3)
lblPc.Caption = info
info = rec.Fields(4)
MsgBox ("Volgende")
rec.MoveNext
Next A
lblUitgevers.Caption = aantalRecords
rec.Close
End Sub
```

Merk op dat ik handig gebruik heb gemaakt van de **aantalRecords** variabele om het maximum te bepalen!

☼ Pas je code aan en probeer!

Veldnamen gebruiken

VBA maakt het ons nog gemakkelijker door ons toe te staan veldnamen te gebruiken, in plaats van de velden te nummeren. In plaats van rec.Fields(1) mogen we dus ook zeggen rec.Fields("Uitgever") en voor rec.Fields(2), rec.Fields("Adres") etc.

Opdracht

☼ Pas je code aan met veldnamen en probeer of het nog goed werkt.

Sessie 2 - 2

Keuzelijst toevoegen.

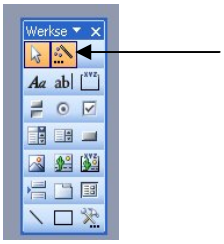
We gaan onze code wat verder aanpassen en willen eigenlijk uit een lijstje een uitgever kunnen kiezen, waarvan we dan de adresgegevens gaan tonen.

Er zijn twee soorten keuzelijst beschikbaar: ***Keuzelijst met invoervak*** en een gewone ***Keuzelijst.*** In dit voorbeeld zie je de verschillen al wel:

De eerste lijst heeft een invoervak. Je kunt daarin typen en dan vliegt de selectie direct naar een item toe dat begint met de letters die je al ingetypt hebt. Ik typte daar "u" in en kreeg direct al "Uithoek" te zien. Door op de lijstknop te klikken krijg je de hele lijst (of deel ervan te zien). Bij een gewone keuzelijst kun je aangeven hoeveel je direct al wilt zien, gewoon door het kader hoger te maken. Als alle items niet direct zichtbaar zijn krijg je schuifbalken om verder te navigeren.
Keuzelijsten met invoervak zijn vaak de beste keuze omdat ze weinig ruimte innemen en je (de eerste letters van) een gezocht item in laten typen.

Als je de wizard aanzet, kun je het te tonen lijstje al direct uit een tabel halen.

☼ Ga na de ontwerpweergave van **frmUitgevers** en zorg ervoor dat de wizard aan staat.



☼ Verwijder de label die het aantal toont.

☼ Plaats een keuzelijst met invoervak en volg de wizard. Kies de

tabel tblUitgevers en daarna het veld Uitgever. Laat de lijst op Uitgever alfabetiseren. Noem de lijst *Uitgever* of *Kies uitgever:* als je het duidelijker wilt zeggen (dit wordt het bijschrift).

☼ Zorg ervoor dat bijschrift en keuzelijst een beetje netjes naast of onder elkaar staan.

☼ Controleer of je kunt kiezen.

Dan gaan we nu onze code aanpassen zodat het gekozen item in de tabel gevonden wordt en het adres wordt getoond.

☼ Ga naar de eigenschappen van de keuzelijst en geef de lijst de naam **kzlUitgever**

☼ Kies uit de tabjes in dit venster de tab **Gebeurtenis**.

☼ Selecteer ***Bij Klikken*** en klik dan op de drie puntjes rechts (…)

☼ Kies hier voor ***Opbouwfunctie voor programmacode***

Je krijgt:

```
Private Sub kzlUitgever_Click()

End Sub
```

Sessie 2 - 3

Private, public en events.

Hier kunnen we nu onze code intypen. Voordat we dat doen, nog een paar opmerkingen. Dit soort codes staan in stukjes die subroutines of procedures heten. *Subroutines* is de naam uit de oude doos, het woordje Sub in de code herinnert daar nog aan. Maar procedure is nu de gebruikelijke term. De procedure wordt hier ingeleid met het woord Private. Dit betekent dat de procedure het privé eigendom is van het formulier dat daar bij hoort. Dit betekent dat andere formulieren niet even gemakkelijk van die procedures gebruik kunnen maken. Het tegenovergestelde van Private is Public. Dan heb je het over code die door jan en alleman (alle formulieren) gebruikt mag worden. Wat dit in de praktijk inhoudt komen we nog wel tegen.

Verder zijn we de term "gebeurtenis" regelmatig tegengekomen. In het Engels heet dat dan "event". Een event kan van alles zijn dat er tijdens het uitvoeren van een programma kan gebeuren en waarop je als programmeur kunt inspelen. Denk daarbij aan het klikken op een knop, het sluiten van een formulier, het maken van een keuze….

Keuzes maken.

Zodra we in de kzlUitgever-lijst op een keuze klikken, kunnen we daar iets mee doen.

We gaan die keuze eerst maar eens opbergen in een variabele en om te controleren of dat wel goed gaat, laten we de Messagebox de inhoud van die variabele even tonen.

☼ Pas de code maar aan en probeer:

```
Private Sub kzlUitgever_Click()
keuze = me.kzlUitgever.Column(1)
MsgBox (keuze)
End Sub
```

Ok, de gegevens staan immers in de tweede (!) kolom, de eerste bevat de Id. Maar nu het juiste adres! Daarvoor moeten we er in de eerste plaats voor zorgen dat de procedure cmdUitgever_Click weet dat er naar de tekst die er in de variabele ***Keuze*** zit gezocht moet worden. Die informatie in ***Keuze*** is ook privé en andere procedures kunnen niet weten wat er in zit. Eigenlijk hebben we ook geen klik op een opdrachtknop meer nodig, maar de code die we ervoor geschreven hebben is nog goed te gebruiken. Dat eerst even regelen.

☼ Kopieer dat relevante stuk uit die opdracht-procdure maar naar de kzl-procedure, je krijgt dan:

```
Private Sub kzlUitgever_Click()
Dim keuze as string
keuze = me.kzlUitgever.column(2)
Dim db As Database
Dim rec As Recordset
Dim aantalRecords As Integer
Set db = CurrentDb()
Set rec = db.OpenRecordset("tblUitgevers")
rec.MoveLast
aantalRecords = rec.RecordCount
rec.MoveFirst
For A = 1 To aantalRecords
info = rec.Fields(1)
lblNaam.Caption = info
info = rec.Fields(2)
lblAdres.Caption = info
info = rec.Fields(3)
lblPc.Caption = info
info = rec.Fields(4)
MsgBox ("Volgende")
rec.MoveNext
Next A
rec.Close
End Sub
```

☼ Probeer (er verandert nog niets!)

Wat nu nog moeten doen is kijken wanneer we de de juiste record aangeboord hebben. We moeten dus kijken naar wat er in de uitgever-kolom zit en zodra dat hetzelfde is als onze keuze, moeten we de bijbehorende gegevens gaan tonen

De constructie die je daarvoor gebruikt heet IF …. Then en ziet er zo uit:

```
If  info = keuze Then
Doe wat er moet gebeuren als het waar is…..
End If
Ga verder…
```

☼ Pas je code als volgt aan en probeer (ik heb de msgbox er uitgegooid want die hebben we niet meer nodig).

```
Private Sub kzlUitgever_Click()
Dim keuze as string
keuze = me.kzlUitgever.column(2)
Dim db As Database
Dim rec As Recordset
Dim aantalRecords As Integer
Set db = CurrentDb()
Set rec = db.OpenRecordset("tblUitgevers")
rec.MoveLast
aantalRecords = rec.RecordCount
rec.MoveFirst
For A = 1 To aantalRecords
info = rec.Fields(1)
if info = keuze then
lblNaam.Caption = info
info = rec.Fields(2)
lblAdres.Caption = info
info = rec.Fields(3)
lblPc.Caption = info
info = rec.Fields(4)
end if
rec.MoveNext
Next A
rec.Close
End Sub
```

☼ Probeer!

We moeten nog wel wat aanpassen, want onze code is nog een beetje te actief en weet niet van ophouden zodra dat wel mag. Immers, zodra het juiste adres gevonden is en weergegeven, blijft de code gewoon doordraaien tot alle records doorzocht zijn. Je past dat aan door de opdracht *Exit For* in te lassen nadat de adresgegevens getoond zijn.

☼ Pas als volgt aan en probeer.

```
…….
info = rec.Fields("Plaats")
lblPlaats.Caption = info
```

```
Exit For
End If
```

Je zult geen verschil merken, maar bij tabellen die duizenden records bevatten kan dat wel het geval zijn.

Sessie 2 – 4

Procedures met elkaar laten communiceren.

Net programmeren houdt in dat je niet teveel door elkaar haalt en procedures alleen dat laat doen waarvoor ze bedoeld zijn. De kzl-procedure bijvoorbeeld was bedoeld om een keuze te maken en niet zozeer om de bijbehorende gegevens uit de tabel te halen en te tonen. We gaan het geheel dan ook eens anders opzetten.

☼ Ga de procedure **kzlUitgever_click** maar eens opdelen in twee procedures. De tweede procedure (het tweede deel) krijgt een zelfbedachte naam, hij is dus ook niet gebaseerd op een gebeurtenis. Het begin en eind van zo'n zelfbedachte procedure moet je dus ook zelf intypen!

```
Private Sub kzlUitgever_Click()
Dim keuze as string
keuze = me.kzlUitgever.column(1)
Toon_gegevens keuze
End sub

Private Sub Toon_gegevens (keuze)
Dim db As Database
Dim rec As Recordset
Dim aantalRecords As Integer
Set db = CurrentDb()
Set rec = db.OpenRecordset("tblUitgevers")
rec.MoveLast
aantalRecords = rec.RecordCount
rec.MoveFirst
For A = 1 To aantalRecords
info = rec.Fields(1)
if info = keuze then
lblNaam.Caption = info
info = rec.Fields(2)
lblAdres.Caption = info
info = rec.Fields(3)
lblPc.Caption = info
info = rec.Fields(4)
end if
rec.MoveNext
Next A
rec.Close
```

```
End Sub
```

Wat gebeurt er hier? De waarde die in de variabele keuze zit wordt met de opdracht

*toon_gegevens **keuze***

naar de procedure *toon_gegevens* gestuurd. De procedure *toon_gegevens* heeft daar al een opvangvariabele (tussen haken) voor klaar staan:

*private sub toon_gevens (**keuze**)*

Hij heet hier ook keuze, maar dat is niet persé nodig. toon_gegevens (mijnkeus) o.i.d. had ook gekund. Het is vaak wel handiger om dezelfde naam te blijven gebruiken.

Zodra je dus een keuze gemaakt hebt, wordt die doorgstuurd naar een aparte procedure die voor de afwerking zorgt.

☼ Probeer.

Omdat we nu die opdrachtknop en de bijbehorende code niet meer nodig hebben kun je die wel verwijderen.

☼ Verwijder de knop en de bijbehorende code. Sluit **boekeniet.mdb.**

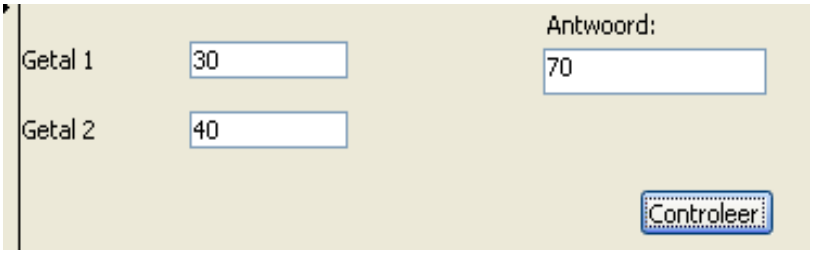
Sessie 2 - 5

We gaan de opgedane kennis nog eens toepassen in ons testformulier.

☼ Open je oefendatabase **OefenDb**

☼ Verwijder het antwoord label en vervang door een tekstveld. Noem dit **txtAntwoord**. Zet "Controleer!" op de TelOp-knop

Je formulier ziet er dan ongeveer zo uit:



☼ Pas nu de code als volgt aan:

```
Private Sub cmdTelOp_Click()
Dim getal1 As Integer
Dim getal2 As Integer
Dim antwoord As Integer
Dim som as Integer
txtGetal1.SetFocus
getal1 = txtGetal1.Text
txtGetal2.SetFocus
getal2 = txtGetal2.Text
som = getal1 + getal2
txtAntwoord.SetFocus
antwoord = txtAntwoord.Text
If antwoord = som Then
MsgBox ("Goed gedaan!")
End If
End Sub
```

☼ Controleer of je de code begrijpt en voer dan uit. Vul twee getallen in en speel dan even met goede en foute antwoorden. Bij een fout antwoord krijg je niets te zien.

Dat is nou niet leuk dat je nooit op je fouten gewezen wordt. Zo iets fantastisch willen we ons zelf niet onthouden. Maar hoe doe je dat dan?

De If..Then constructie kan nog wat aangepast worden en leest dan:

```
If   (vergelijk iets) Then  (als het waar is)
Doe iets
Else (als het niet waar is)
Doe iets anders
End IF
```

☼ Pas de code als volgt aan en probeer!

```
Private Sub cmdTelOp_Click()
Dim getal1 As Integer
Dim getal2 As Integer
Dim antwoord As Integer
txtGetal1.SetFocus
getal1 = txtGetal1.Text
txtGetal2.SetFocus
getal2 = txtGetal2.Text
txtAntwoord.SetFocus
antwoord = txtAntwoord.Text
If antwoord = getal1 + getal2 Then
MsgBox ("Goed gedaan!")
Else
MsgBox("Foutje!")
End If
```

```
End Sub
```

We gaan dit nog even wat interessanter maken door Access zelf getallen te laten kiezen.

☼ Plaats een opdrachtknop **cmdNieuw** (met de tekst **Nieuw** erop). (Klik op Annuleer als de Wizard start of schakel de Wizard van te voren uit)

☼ Voer de volgende procedure in en bewaar daarna je formulier.

```
Private Sub cmdNieuw_Click()
Dim hoogste as integer
Dim getal1 as integer
Dim getal2 as integer
Randomize
Hoogste = 100
getal1 = Int(Hoogste * Rnd) + 1
getal2 = Int(Hoogste * Rnd) + 1
txtGetal1.SetFocus
txtGetal1.Text = getal1
txtGetal2.SetFocus
txtGetal2.Text = getal2
End Sub
```

De functie **Rnd** genereert een getal tussen 0 en 0,9999999999…
Door de procedure te laten voorafgaan door **Randomize** wordt iedere keer een volledig nieuwe set willekeurige getallen gegenereerd. **Randomize** bepaalt het startgetal op basis van de interne klok (datum+tijd).

Stel dat je een getal wilt hebben tussen 0 en 6 dan vermenigvuldig je het willekeurige getal met 6: 6 * Rnd Dit geeft je een getal tussen 0 en 5,99999999….. Als je er vervolgens voor zorgt dat alles achter de komma verdwijnt, krijg je de getallen 0 t/m 5. Je doet dat met de Int-functie Int(6 * Rnd). Mocht je liever 1 t/m 6 hebben, dan hoef je er alleen nog maar 1 bij op te tellen: **Int(6 * Rnd) + 1** Zo komen we dus aan onze willekeurige getallen.

☼ Probeer.

Om het je zelf nog moeilijker te maken, zou je de getallen ook kunnen laten vermenigvuldigen:

☼ **Pas aan:** If antwoord = getal1 * getal2 Then **(* i.p.v. +)**

☼ Probeer als je het aandurft.

Sessie 2 - 6

Oefening 1

☼ Voeg een tekstveld toe waarin je het hoogste getal bepaalt (i.p.v. 100) en pas je code aan zodat dat getal als hoogste getal gebruikt gaat worden.

Sessie 2 - 7

Oefening 2 (Hoofdstedenspel)

- ☼ Sluit de OefenDB-database en open Boekenier weer.
- ☼ Importeer de gegevens uit **Hoofdsteden.xls** in je access database (laat Access de primaire sleutel toevoegen) Je kunt de Excel werkmap vinden op de cursus website.
- ☼ Zet een nieuw formulier op, noem het maar **tblHoofdsteden**.
- ☼ Plaats een Keuzelijst met Invoervak (**kzlHoofdsteden**) dat een lijst geeft van alle Hoofdsteden.
- ☼ Plaats een label **lblLand** en een **lblHoofdstad**
- ☼ Plaats een opdrachtknop **cmdVolgende**.

Zoiets:

Wat we nu gaan doen is Access een land laten kiezen (als je op **Volgende** klikt) en tonen. Jij mag dan de hoofdstad erbij halen uit de keuzelijst. Dit laat je dan op correctheid controleren, met de uitslag natuurlijk ("Goed" of "Fout").

Instructies:

Met de opdracht *rec.move waarde* kun je naar de record gaan die in de variabele *waarde* zit. Zit daar bijv. 6 in dan ga je naar de zesde record (vanaf waar je bent!) i.p.v. bijv. de volgende (wat *rec.MoveNext* doet). Door ervoor te zorgen dat de variabele *waarde* een willekeurig getal komt te herbergen, heb je zo een kennisspelletje gemaakt.

Stappen:

Voor de **cmdVolgende** procedure.

1. Open en initialiseer de database en de correcte tabel.
2. *Move* naar de laatste record, tel het aantal records en move weer terug naar de eerste. (Dit is een standaard procedure om zeker te zijn dat alles werkt)

3. Genereer een willekeurig getal met als hoogste het aantal records – 1 (want de recordnummering begint bij 0!) Hier moet je even goed over nadenken. Je hebt de getallen 0 t/m ? nodig.
4. Plaats dat in *waarde* of welke variabele je ook maar wilt gebruiken.
5. Gebruik *rec.move waarde* om die record (vanaf het begin) te benaderen.
6. Plaats het land in *lblLand*
7. Maak *lblHoofdstad* onzichtbaar (met een eigenschap van dit label!)
8. Plaats de hoofdstad in *lblHoofdstad*

Voor de kzlHoofdstad procedure.

1. Plaats de keuze in een variabele.
2. Vergelijk de keuze met de inhoud van *lblHoofdstad*
3. Is het correct, toon de boodschap "goed!" of iets dergelijks, anders "fout!" of zoiets.
4. Maak *lblHoofdstad* weer zichtbaar.

☼ Sluit de database zodra het werkt en je er genoeg van hebt.

Sessie 2 - 8

Opdracht 3

☼ Start een nieuw formulier in **boekenier.mdb** , met dezelfde opmaak als **frmUitgevers** door **frmUitgevers** te kopieren en weer te plakken. Noem de nieuwe tabel **frmKlanten**.

☼ Pas dit formulier aan zodat het voor klanten geldt.

☼ Verwijder de keuzelijst en plaats hem opnieuw (met de wizard aan) gebaseerd op **tblKlanten**. Voeg **achternaam, titel, voorletters, voorvoegsels toe**.

☼ Zorg ervoor dat naam **kzlKlanten** is en er een klik-gegbeurtenis aan gekoppeld wordt.

☼ Ga naar je vbacode en kopieer de code uit de **kzlUitgevers**_click-procedure naar **kzlKlanten**_click. Verwijder daarna de **kzlUitgevers**-procedure. Pas die code aan zodat het het ID uit de keuze haalt.

☼ Pas nu de opzoekprocedure uit. Je controleert op ID en moet dus code toevoegen voor *Achternaam, titel, voorletters en voorvoegsels*. Verzamel al die gegevens in variabelen.

☼ Zorg ervoor dat al die naamgegevens in één variabele terecht komen. Je doet dat als volgt:

` Verzamelvariabele = Var1 & " " & Var2 & " " & etc.

& wordt gebruikt om teksten aan elkaar te plakken, maar omdat je er ook wat ruimte tussen wilt, voeg je spaties toe. Spaties codeer je als " "

☼ Kijk ook even goed of de rest wel goed is, zijn de veldnamen in **tblKlanten** helemaal gelijk aan die in **tblUitgevers**??

☼ Probeer dan en zie of je een naam kunt kiezen en het juiste adres opgeleverd krijgt.

Succes!

Derde ronde

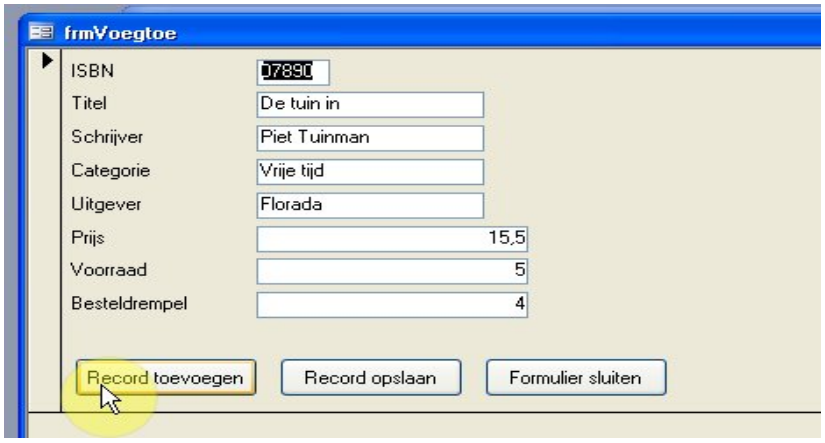
Sessie 3 - 1

In deze ronde beginnen we met het maken van een toevoegformulier voor nieuwe boeken. Daarvoor moeten we toevoegen aan de ISBN-tabel en dan kunnen we ook gelijk de voorraadtabel wel meenemen.

Zorg ervoor (als je dat al niet gedaan hebt) dat je velden niet meer tekens innemen dan nodig zijn. Doe je dat niet dan gaat de wizard grotere tekstvakken produceren dan nodig en mooi is.

☼ Pas de tekenbreedte van je velden in de tekstvelden van tblISBN aan.

☼ Maak een formulier (**frmVoegtoe**) gebaseerd op tblISBN en tblVoorraad (laat het voorraadISBN-nummer weg). Voeg 3 knoppen toe (cmdToevoegen/cmdOpslaan/cmdSluiten) die een nieuw record toevoegen, het record opslaan en het formulier sluiten. Het moet zoiets worden:



☼ Voeg toe:

tblISBN

ISBN	Titel	Schrijver	Categorie	Uitgever	Prijs
07900	Het veld uit	Piet Tuinman	Vrije tijd	Florada	19,5
07901	De Vrijgezel	Flier Fluiter	Roman	Florada	16,5

Voorraad	Besteldrempel
5	3
5	3

☼ Sluit het formulier en zie of beide tabellen correct aangevuld zijn. Sluit de tabellen daarna weer.

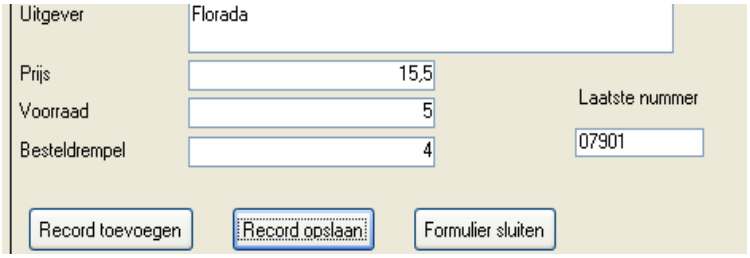
Sessie 3 - 2

ISBN-nummers krijg je toegewezen, maar voor onze winkel gaan we net doen of we deze codes zelf toe moeten kennen. We pakken telkens gewoon het volgende nummer (met een verhogen dus) en daarom zou het wel handig te zijn om direct te zien welke het laatst gebruikt werd.

Hoe zou je dat toen?

☼ Plaats een veld en zet daarin een functie die je het hoogste tot nu toe gebruikte nummer geeft. (bekijk de video nogmaals als je er niet uitkomt)

Dit zou het resultaat moeten zijn:



Sessie 3 - 3

Het invullen van dit formulier zou gemakkelijker moeten kunnen. Je zou categorie uit een lijstje aan moeten kunnen klikken en eigenlijk zou dat voor schrijver en uitgever ook moeten kunnen. De lijstjes moeten dan wel up-to-date blijven, zodat nieuwe schrijvers en eventueel categorieën telkens automatisch worden toegevoegd. Voor uitgevers hebben we al een tabel. Darvoor moeten we nog wel een toevoegformulier maken.

We gaan dit als volgt oplossen:

1. Maak tabelmaakqueries die tabellen verzorgen van de gewenste gegevens (**tblCategorie/tblSchrijvers**)
2. Pas **tblIsbn** aan zodat er uit lijsten gekozen kan worden (voor uitgevers uit de tabel **tblUitgevers**)
3. Pas **frmVoegtoe** aan zodat de velden op basis van de nieuwe tabelvelden aangepast worden.
4. Maak een toevoegformulier voor Uitgevers (met de benodigde knoppen)

Daar gaat ie dan:

☼ Maak een tabelmaakquery gebaseerd op **tblIsbn** met alleen maar de **Schrijvers**. Zorg ervoor dat er gegroepeerd wordt, waardoor je alle schrijvers maar 1 keer krijgt. Noem de nieuw te maken tabel **tblSchrijvers**. Voer de query uit en verwijder hem dan maar weer (is niet meer nodig). Controleer of de tabel gemaakt is.

☼ Doe hetzelfde voor de categorieën, noem de nieuwe tabel maar **tblCategorie**.

☼ Pas de velden **Schrijver** en **Categorie** in de tabel(!) **tblIsbn** aan zodat de **wizard-opzoeken** ze verbindt aan de bijbehorende gegevenstabellen.

☼ Maak een toevoegformulier (in **Tabelvorm**) voor de uitgevers (pas eventueel de veldlengtes eerst aan), noem het maar **frmMutatieUitgevers**. Voeg ook een **Sluit**knop toe.

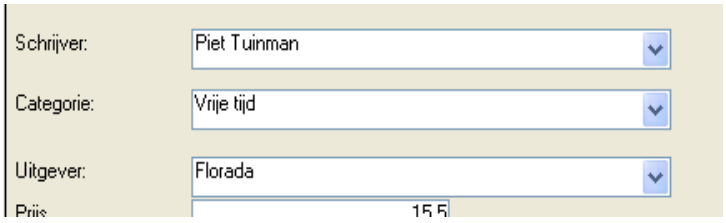
☼ Pas het veld **Uitgevers** in de tabel tblIsbn aan zodat er geselecteerd kan worden uit de **tblUitgevers** (je gaat hier alle informatie in eerste instantie verliezen)

☼ Vul de uitgevergegevens weer in met behulp van het selectieknopje:

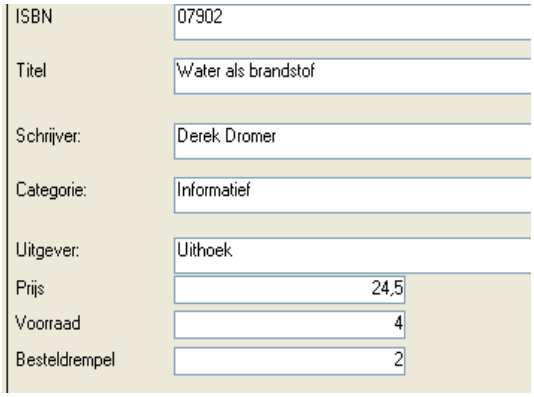
☼ Pas nu het formulier aan:

Verwijder de oorspronkelijke velden en plaats dezelfde velden opnieuw. Je krijgt de velden te zien met behulp van het ***lijst met velden*** *- knopje*

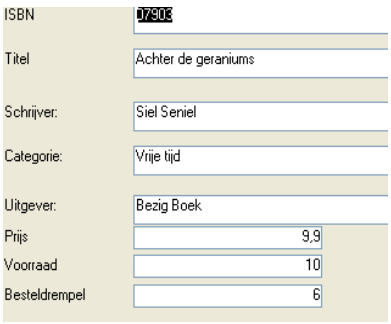
Je krijgt nu selectieknoppen te zien in je velden:



☼ Voeg nog maar even een titel toe en maak gebruik van de selectievelden:



Ok, al het basiswerk achter de rug. Maar nu het onderhoud.



☼ Voeg weer eens een nieuwe titel toe in **frmVoegtoe** (met een nieuwe schrijver, die je dus in moet typen):

☼ Sluit het formulier en kijk in **tblSchrijvers**. De nieuwe schrijver werd niet automatisch toegevoegd.

Sessie 3 - 4

Om dit wel voor elkaar te krijgen moeten die tabellen (tblSchrijvers en tblCategorie) ge-update worden. Dat updaten zou je kunnen doen door zo'n tabel eerst te legen en dan weer te vullen.

☼ Maak verwijderqueries voor **tblSchrijvers** en **tblCategorieën**. Noem ze maar **qrySchrijversLeeg** en **qryCategorieLeeg**.

Vul nu de tabellen weer met de gegevens uit **tblIsbn** als volgt:

☼ Maak toevoegqueries gebaseerd op **tblIsbn** die de tabellen **tblSchrijvers** en **tblCategorie** weer vullen met de gewenste gegevens. Noem ze maar **qrySchrijversToe** en **qryCategorieToe**.

We gaan er nu voor zorgen dat dit telkens automatisch gebeurt, bijv bij het opslaan van het record.

We doen dit eerst even met een testknop.

☼ Voeg een testknop toe aan je formulier frmVoegtoe (cmdVernieuw) en ga dan naar de bijbehorende click-code:

```
Private Sub cmdVernieuw_Click()

End Sub
```

In deze routine gaan we de al eerder gemaakte queries uit laten voeren.

☼ Pas als volgt aan:

```
Private Sub cmdTest_Click()
DoCmd.OpenQuery "qrySchrijversleeg", , acReadOnly
DoCmd.OpenQuery "qrySchrijversToe", , acReadOnly
DoCmd.OpenQuery "qryCategorieleeg", , acReadOnly
```

```
DoCmd.OpenQuery "qryCategorieToe", , acReadOnly
End Sub
```

De opdracht DoCmd. heeft veel mogelijkheden om van alles in Access VBA gedaan te krijgen. We gebruiken hem hier om een query uit te laten voeren. Hij bestaat uit 3 "parameters". De eerst is de naam van de query, de tweede die hier leeg is (staat tussen de twee komma's (, ,) kun je eventueel vullen met de code *acViewDesign*. Dan krijg je bij het uitvoeren eerst de ontwerpweergave te zien. Standaard gebeurt dit niet en dat wil je waarschijnlijk ook niet. De laatste parameter (acReadOnly) geeft aan dat er verder niets veranderd mag worden. Bij het uitvoeren krijg je allerlei waarschuwingen te zien. Die gaan we straks uitschakelen.

☼ Voer uit (eerst even zonder nieuwe gegevens, om te zien of de code werkt)!

☼ Voeg nieuwe gegevens toe:

ISBN	Titel	Schrijver	Categorie	Uitgever	Prijs	Voorraad	Besteldrempel
07904	Op goed geluk	Maarten Mazzel	Roman	Bezig Boek	24,45	5	3

☼ Klik op **Vernieuw** en start een nieuwe record

(Maarten moet nu ook in het Schrijverslijstje staan)

Om niet met die waarschuwingen lastig gevallen te worden, moet je access dat vertellen.

☼ Pas je code als volgt aan:

```
Private Sub cmdVernieuw_Click()
DoCmd.SetWarnings False
DoCmd.OpenQuery "qrySchrijversleeg", , acReadOnly
DoCmd.OpenQuery "qrySchrijversToe", , acReadOnly
DoCmd.OpenQuery "qryCategorieleeg", , acReadOnly
DoCmd.OpenQuery "qryCategorieToe", , acReadOnly
Me.Requery
Me.Refresh
End Sub
```

Het op **False** zetten van de waarschuwingen, heeft wel het gevolg dat een aantal inwendige automatische acties die bij die waarschuwingen horen, ook niet meer uitgevoerd worden. Het formulier moet de bijbehorende gegevens weer opnieuw opvragen (**me**.**Requery**) en weer vernieuwd worden (**me**.**Refresh**). "Me" verwijst naar het formulier waar de code aan vast zit.

Pas je code als volgt aan:

☼ Probeer met weer een nieuw setje gegevens:

	ISBN	Titel	Schrijver	Categorie	Uitgever	Prijs	Voorraad	Besteldrempel
▶	07905	Kopen zonder kosten	Gerrie Gratis	Informatief	Valuta	49,5	3	2
*								

Omdat we eigenlijk liever niet telkens op die Vernieuw-knop willen hoeven te klikken, gaan we de nu uitgetest functie onder één van die andere knoppen hangen. De Opslaan-knop, wellicht, want daarna hebben we de nieuwe gegevens weer nodig.

☼ Zoek de **cmdOpslaan_click()** procedure.

☼ Kopieer de code uit **cmdVernieuw** naar deze procedure en plaats ze net voor de **Exit_cmdOpslaan_Click**: regel:

```
Private Sub cmdOpslaan_Click()
On Error GoTo Err_cmdOpslaan_Click


    DoCmd.DoMenuItem acFormBar, acRecordsMenu
DoCmd.SetWarnings False
DoCmd.OpenQuery "qrySchrijversLeeg"
DoCmd.OpenQuery "qrySchrijversToe"
DoCmd.OpenQuery "qryCategorieLeeg"
DoCmd.OpenQuery "qryCategorieToe"
Me.Requery
Me.Refresh
Exit_cmdOpslaan_Click:
    Exit Sub

Err_cmdOpslaan_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpslaan_Click

End Sub
```

☼ Probeer met maar weer een nieuwe titel:

	ISBN	Titel	Schrijver	Categorie	Uitgever	Prijs	Voorraad	Besteldrempel
▶	07906	Dagje Ameland	Wander Wandelaar	Reizen	Uithoek	8,5	10	5

Sessie 3-5

Locale en globale variabelen

☼ Open de database **OefenDB**

☼ Open frmHoofdsteden in Ontwerpmodus en voeg een label toe lblCijfer.

☼ Type **Cijfer** in het label en zet de lettergrootte op 24

Als je een waarde aan een variabele toekent dan onthoudt hij dat alleen maar tijdens de loop van een procedure.

☼ Ga naar de code van frmHoofdsteden en pas de code van **kzlHoofdsteden_click** als volgt aan:

```
Private Sub kzlHoofdsteden_Click()
dim aantalgoed as integer
keuze = kzlHoofdsteden.Column(1)
If lblHoofdstad.Caption = keuze Then
MsgBox ("Goed!")
aantalgoed = aantalgoed + 1
Else
MsgBox ("Fout!")
End If
lblCijfer.Caption = aantalgoed
lblHoofdstad.Visible = True
End Sub
```

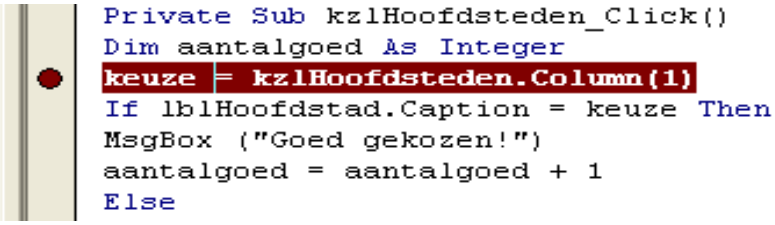
De code *aantalgoed* = *aantalgoed* + 1 zou telkens 1 moeten toevoegen aan *aantalgoed*, maar omdat aantalgoed de waarde niet vastgehouden heeft als de procedure weer opnieuw start, zal het niet veel opschieten.

☼ Probeer maar.

Code volgen

Om te zien wat er gebeurt in onze code kun je stap voor stap door de code lopen.

☼ Ga terug naar je code en plaats de cursor in de regel
Keuze = kzlHoofdsteden.Column(1)



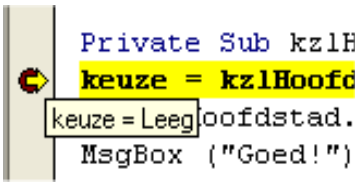
☼ Druk nu op de functietoets F9 (de regel krijgt een bruine kleur)

☼ Ga terug naar het formulier, start het spel en kies je eerste hoofdstad.

Het programma stopt bij de aangegeven regel. De regel is nu grotendeels geel.

☼ Plaats je cursor op **keuze**

Je ziet nu dat **keuze** nog leeg is.



☼ Druk op de functietoets F8 (de volgende regel is nu geel)

☼ Plaats je cursor weer op **keuze**. (**keuze** is nu wel gevuld met een naam)

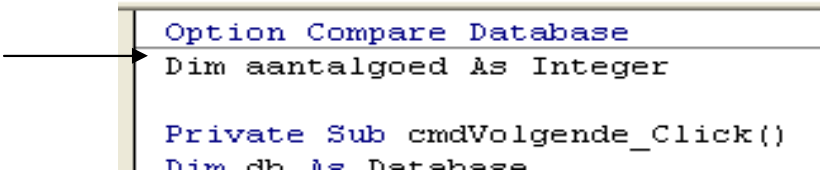
☼ Ga verder met F8 en let eens op hoe **aantalgoed** eerst 0 bevat en daarna 1 (je moet de boodschap telkens even OK-en)

☼ Loop verder door de routine tot je aan het eind bent (geen geel meer ziet).

☼ Ga nu met **Volgende** enz. naar de volgende vraag. Loop de code weer door en zie hoe **aantalgoed** weer opnieuw met 0 gevuld is aan het begin.

Zodra je de routine verlaten hebt, wordt zo'n variabele weer vrijgegeven. Variabelen die alleen maar binnen de procedure hun waarden behouden heten locale variabelen. Maar voor wie wat verder denkt dan de eigen buurt zijn er ook globale variabelen, variabelen die hun waarde behouden voor alle procedures op zo'n formulier. Zulke variabelen moet je dan ook los van die specifieke procedures declareren. Je doet dat helemaal bovenaan in je code venster.

1. Pas maar aan:



Dat Option Compare Database bovenaan bepaalt de manier waarop Access sorteert, alfabetiseert. We laten dat voorlopig even voor wat het is.

☼ Verwijder de oorspronkelijke locale declaratie.

2. Probeer nu nog maar eens en volg de code.

3. Plaats de cursor weer in de bruine regel en druk nogmaals op F9

De "stop" moet nu weer verdwenen zijn.

☼ Probeer.

Sessie 3 – 6

Oefening 1

Eigenlijk wou ik liever bijhouden hoeveel er goed en hoeveel er fout gekozen worden en daarna een cijfer toekennen.

☼ Aanwijzingen:

1. Gebruik de globale variabelen **aantalgoed** en **aantalfout**.
2. Laat ze beide tellen.
3. Bedenk een formule die een cijfer toekent van 0 tot 10.

4. Plaats dit cijfer telkens in lblCijfer
5. Probeer eens uit te vinden welke eigenschap wordt gebruikt om de letters in een label een kleurtje te geven (heb je de eigenschap gevonden dan kun je met de drie puntjes … de kleuren kiezen). Onthoud het nummer voor rood en groen.
6. Pas de code aan zodat de letters rood worden bij een cijfer lager dan 6 en groen bij 6 t/m 10.

☼ Probeer.

Vierde ronde

Sessie 4 – 1

Recordset baseren op een query

We gaan ervoor zorgen dat alle goed geraden hoofdsteden met hun landen verdwijnen uit het te raden setje.

Doel: De goedgeraden hoofdsteden met hun landen niet weer tonen.

Hier zijn de benodigde stappen:

1. Markeren van de goedgeraden hoofdsteden.
2. Ervoor zorgen dat alleen de nog niet goed geraden hoofdsteden aangeboden worden.
3. De lijst van hoofdsteden (de stedenlijst) telkens vullen met alleen de nog niet goed geraden steden.
4. Ervoor zorgen dat aan het begin alles weer op standaard wordt ingesteld

Hoe doe je dat?

5. Toevoegen van een **ja/nee**-veld aan de **tblHoofdsteden** en dat veld op **Waar**(**True**) laten zetten wanneer een hoofdstad goed geraden werd.
6. Het aanbod baseren op een query met alle nog niet goedgeraden steden.
7. De stedenlijst legen en weer vullen met de nog niet goedgeraden steden.
8. Alle als goed geraden gemarkeerde records weer herstellen als nog niet geraden bij opnieuw spelen.

En nu aan het werk:

☼ Voeg een veld toe aan de tblHoofdsteden met als **type ja/nee.** Noem hem maar **Correct.**

☼ Maak een query gebaseerd op **tblHoofdsteden** die alle gegevens toont waarvoor Correct False is. Bewaar als **qryHoofdsteden.**

☼ Pas de code nu aan als volgt: (merk op dat ik overal **rec** vervangen heb door **qrec)**

```
Private Sub cmdVolgende_Click()
Dim db As Database
Dim qrec As Recordset
Dim aantalRecords As Integer
Set db = CurrentDb()
Set qrec = db.OpenRecordset("qryHoofdsteden")
qrec.MoveLast
aantalRecords = qrec.RecordCount
qrec.MoveFirst
Randomize
```

```
waarde = Int(aantalRecords * Rnd)
qrec.Move waarde
lblLand.Caption = qrec.Fields(1)
lblHoofdstad.Visible = False
lblHoofdstad.Caption = qrec.Fields(2)
qrec.Close
End Sub
```

Het vervangen van van **rec** door **qrec** is een persoonlijke voorkeur. Het is niet perse nodig, maar het helpt mij om te zien dat deze records door een query zijn gegenereerd en niet direct uit een tabel komen.

Wat we nu nog moeten doen is **Correct** op **True** zetten zodra een hoofdstad goed geraden is.

We gaan dit in een volledig nieuwe procedure doen, die we dan vanuit de keuzeprocedure aan gaan roepen. Ik gebruik de definitie Private maar niet, want zo privé is die procedure nou ook weer niet.

Records bewerken

☼ Schrijf maar (kopieer een stuk uit de eerdere procedure!)

```
Sub markeer()
Dim db As Database
Dim qrec As Recordset
Dim aantalRecords As Integer
Set db = CurrentDb()
Set qrec = db.OpenRecordset("qryHoofdsteden")
qrec.MoveLast
aantalRecords = qrec.RecordCount
qrec.MoveFirst
qrec.Move waarde
qrec.Edit
qrec.Fields("correct") = True
qrec.Update
qrec.Close
End Sub
```

Om een waarde te wijzigen moet je Access eerst in de Edit-modus zetten en nadat je een nieuwe waarde aan dat veld hebt toegekend, moet het ook nog eens specifiek ge-update worden. Dit heeft er mee te maken dat de mutatie in een tijdelijk tabelletje plaats vindt, zodat alles nog gemakkelijk teruggedraaid kan worden. De update maakt de wijziging definitief.

☼ Roep deze routine aan vanuit **kzlHoofdsteden_Click**() wanneer het antwoord goed is.

```
Private Sub kzlHoofdsteden_Click()
keuze = kzlHoofdsteden.Column(1)
If lblHoofdstad.Caption = keuze Then
MsgBox ("Goed!")
aantalgoed = aantalgoed + 1
```

```
markeer
Else
MsgBox ("Fout!")
aantalfout = aantalfout + 1
End If
cijfer = Int(aantalgoed / (aantalgoed + aantalfout) * 10 + 0.5)
If cijfer < 6 Then
lblCijfer.ForeColor = 255
Else
lblCijfer.ForeColor = 65280
End If
lblCijfer.Caption = cijfer
lblHoofdstad.Visible = True
End Sub
```

- ☼ Zorg ervoor dat **waarde** een globale variabele is!!!

- ☼ We zijn er nog niet, maar probeer maar eens om (na een goed antwoord) te kijken in **tblHoofdsteden** of dat record ook werkelijk aangevinkt is.

Sessie 4 - 2

Oefening:

- ☼ Maak een verwijderquery die de tabel **tblStedenlijst** wist: **qryStedenlijstLeeg**

- ☼ Maak een toevoegquery die **tblStedenlijst** weer vult maar alleen met de **Onware** gegevens: **qryStedenlijstVul**

- ☼ Laat deze queries door je code uitvoeren (aan het eind van de **markeer**-procedure). Zorg er ook voor dat je niet telkens die waarschuwingen krijgt.

- ☼ Zie of het werkt.

Je moet nu wel telkens alle correct-velden eerst weer op onwaar zetten zodra je het spel opnieuw wilt spelen.

- ☼ Maak een **Bijwerk**query die alle correctveldjes weer herstelt (**qryHoofdstedenherstel**)

- ☼ Plaats een opdrachtknop met de caption **Opnieuw** die de query uitvoert en er ook voor zorgt dat de stedenlijst weer correct is.

Die queries die **tblStedenlijst** legen en weer vullen moet je nu op twee verschillende plaatsen oproepen,

- ☼ Bedenk een manier waarop die code maar één keer ingevoerd hoeft te worden. Noem die extra benodigde procedure maar "**herstel_steden**".

- ☼ Speel en test

Tenslotte zou het ook nog mooi zijn als bij het opnieuw starten van de applicatie, alles ook weer even hersteld wordt. Je kunt dat mooi doen op het moment waarop het spelformulier geladen wordt. Je krijgt dan de zgn laad-gebeurtenis. Je kunt daarop inhaken door in je vba-editor voor het formulier, uit het lijstje links-bovenaan **Form** te kiezen en rechts **Load**.
Het procedureskelet wordt dan al voor je klaar gezet:

```
Private Sub Form_Load

End Sub
```

Daarin kun je dan de gewenste code plaatsen, zoals hierboven:

☼ Probeer dit voor elkaar te krijgen en test het.

Omdat je die code in Form_Load ook voor je **Opnieuw**-knop nodig had, zou je er ook voor kunnen zorgen dat dat door een aparte procedure wordt gedaan, bijv.

```
Sub Herstel
DoCmd.SetWarnings False
DoCmd.OpenQuery ("qryHoofdstedenHerstel")
End Sub
```

☼ Maak hier gebruik van en pas de zaak aan.

Ten slotte zou het nog mooi zijn om ervoor te zorgen dat bij een nieuwe start de labels **lblCijfer**, **lblHoofdstad** en **lblLand** weer hun oorspronkelijke tekst terug krijgen.

☼ Zorg hier voor.

Sessie 4 - 3

Probleem met automatische nummering.

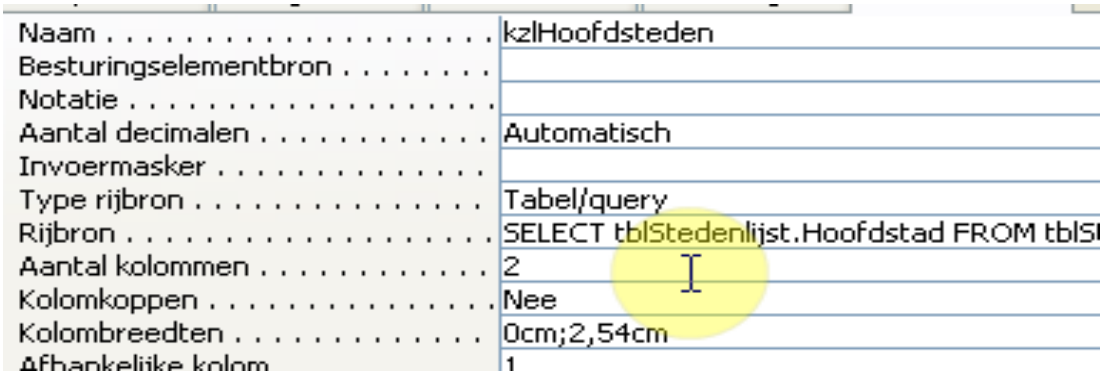
Als je eens naar je **tblStedenlijst** kijkt, zie je dat bij iedere nieuwe vulling, de autonummering gewoon doorloopt waar hij bij de vorige geëindigd was. Niet bepaald wenselijk, want waar houdt dat ooit een keer op. Het gebruik van autonummering hier was dus niet een erg gelukkige keus, bovendien had je het hier niet nodig, daar de hoofdsteden zelf al uniek zijn. Ik heb dat veld dan ook maar weer verwijderd. Maar dan zijn nog wel een aantal verdere aanpassingen nodig. Het wordt in de video voorgedaan.

☼ Verwijder het **Id**-veld (Het hoofdstedenformlier mag niet open zijn)

☼ Ga naar de eigenschappen van **kzlHoofdsteden** en verwijder uit de eigenschap **Rijbron** het stukje **tblStedenlijst.Id**, (inclusief de komma dus).

☼ Kies het tabblad **Alles** (als je dat nog niet gedaan had)

☼ Verander de eigenschap **Kolomkoppen** in 1 en verwijder uit **Kolombreedten** *0cm;*



☼ Verander ten slotte in je code voor **kzlHoofdsteden** ***Column(1)*** in ***Column(0)***

```
Private Sub kzlHoofdsteden_Click()
keuze = kzlHoofdsteden.Column(0)
If lblHoofdstad.Caption = keuze Then
```

☼ Zie of het nog werkt.

Sessie 4 - 4

Fouten (errors)

Programma's kunnen fouten melden zodra de programmaregels niet meer met de gegevens overweg kunnen. Zodra je alle hoofdsteden gehad hebt, is je tabel **tblHoofdsteden** leeg. Probeer je dan bijv. naar de laatste record van die tabel te gaan, bestaat die niet en produceert VBA een foutmelding. Soms kun je proberen potentiële fouten te voorkomen, maar soms geeft het ook niet om ze maar te laten gebeuren en er dan iets mee te doen. Zodra **MoveLast** niet meer werkt, weet je dat je het einde bereikt hebt van de serie records en dat je op kunt houden met inlezen. Zo'n op handen zijnde foutmelding (een Error – event) kun je dan opvangen en afhandelen zonder dat de gebruiker het merkt.

Zodra er geen hoofdsteden over zijn om te kiezen en je klikt toch weer op volgende in **frmHoofdsteden**, zal er dus ook een fout optreden.

☼ Open de Oefendatabase weer (indien niet meer open) en open het formulier frmHoofddteden.

☼ Open het formulier **frmHoofdsteden** (niet weer sluiten).

☼ Open de tabel tblHoofdsteden en markeer alle correct-velden behalve de laatste.

☼ Voer de query's **qryStedenlijstLeeg** en **qryStedenlijstvul** uit

Je hebt nu nog maar één niet geraden hoofdstad over en je stedenlijst is ook aangepast.

☼ Speel het spel en raad de overgebleven hoofdstad, klik daarna op **Volgende**.

Je krijgt de volgende foutmelding.

☼ Klik op **Foutopsporing** en je zit gelijk in je editor.

De gele regel geeft aan waar de fout optrad!

☼ Peuter de foutprocedure los en bekijk de volgende code, let op de toevoegingen.

```
Private Sub cmdVolgende_Click()
Dim db As Database
Dim qrec As Recordset
Dim aantalRecords As Integer
On Error GoTo Foutje
Set db = CurrentDb()
Set qrec = db.OpenRecordset("qryHoofdsteden")
qrec.MoveLast
aantalRecords = qrec.RecordCount
qrec.MoveFirst
Randomize
waarde = Int(aantalRecords * Rnd)
qrec.Move waarde
lblLand.Caption = qrec.Fields(1)
lblHoofdstad.Visible = False
lblHoofdstad.Caption = qrec.Fields(2)
qrec.Close
GoTo Einde
Foutje:
MsgBox ("Voltooid")
Einde:
End Sub
```

On Error GoTo Foutje vertelt het programma dat als er daarna een fout optreedt, het programma direct door moet gaan naar de regel Foutje: Let op, zo'n regel waar je een programma naar toe kunt sturen moet altijd in een dubbele punt eindigen. Je mag zelf weten hoe je die regel noemt: stommiteit: vergissing: etc. als hij maar door : gevolgd wordt.

Als er dus een fout plaats vindt, slaat het programma alles over en gaat naar de genoemde regel en daar verder. In de code hierna, kun je dus mooi vermelden dat er iets fout ging, of als je van die fout handig gebruikt wilt maken, kun je een andere boodschap tonen. Wij hebben geen hoofdsteden meer over, dat was de reden van de fout en we kunnen nu mooi de boodschap tonen dat het spel voorbij is. Wat je echter niet wilt is dat die boodschap telkens getoond wordt, want normaal gesproken zou de code na *qrec.Close* gewoon doorlopen en ook het fout-bericht tonen. Dit vang je op door net voor de foutregel een andere GoTo-code te plaatsen die het programma voorbij de foutafhandeling stuurt. Ik gebruik vaak zoiets als Einde: om het programma naar toe te sturen.

☼ Type in en probeer.

Je mag voor *Goto Einde* wel eens een enkele aanhalingsteken plaatsen, de tekst wordt dan groen en doet niet meer mee: 'Goto Einde

☼ Probeer maar en bekijk het effect. Haal daarna het aanhalingsteken weer weg.

Sessie 4 - 5

Oefening:

Professoren zouden in het telop-programma wel eens heel vergeetachtig er niet aan kunnen denken het hoogst gewenste getal in te typen en zij en minder hoog opgeleiden proberen je programma wellicht te plagen met het intypen van letters i.p.v. getallen, “tien” bijvoorbeeld i.p.v. 10.

☼ Probeer dat maar eens.

☼ Zoek uit waar de fouten zitten (in welke regels). Zorg er dan voor dat de gebruiker bij deze fouten even eenvoudig de boodschap krijgt dat dit zo niet kan.

Fouten voorkomen.

Soms kun je fouten ook gemakkelijk voorkomen. Je zou bijvoorbeeld kunnen controleren of er nog wel records beschikbaar zijn in de tabel die ik **tblStedenlijst** genoemd heb en waarin de nog niet geraden hoofdsteden opgeslagen worden. Als er niets meer inzit dan kan het tellen van de records ook alleen nog maar nul opleveren. Je kunt hierop controleren. Dat moet je wel doen voordat je .**MoveLast** uitvoert anders krijg je nog eerst een fout. Je past je code dan als volgt aan (Ik heb de foutopvang verwijderd):

```
Private Sub cmdVolgende_Click()
Dim db As Database
Dim qrec As Recordset
Dim aantalRecords As Integer
Set db = CurrentDb()
Set qrec = db.OpenRecordset("qryHoofdsteden")
If qrec.RecordCount = 0 Then GoTo Voltooid
qrec.MoveLast
aantalRecords = qrec.RecordCount
If IsNull(qrec.RecordCount) Then GoTo Einde
qrec.MoveFirst
Randomize
waarde = Int(aantalRecords * Rnd)
qrec.Move waarde
lblLand.Caption = qrec.Fields(1)
lblHoofdstad.Visible = False
lblHoofdstad.Caption = qrec.Fields(2)
qrec.Close
GoTo Einde
Voltooid:
MsgBox ("Voltooid")
Einde:
End Sub
```

☼ Pas maar aan en probeer.

Sessie 4 - 6

Tabellen tonen en sluiten.

We kunnen nu query's maken en die vanuit VBA aanroepen en uit laten voeren. Het gaat dan wel om volledig ingevulde query's, waar je verder niets aan kunt veranderen. Het is vaak handig om dat wel te kunnen. Daar gaan we in deze sessie de voorbereidingen voor treffen. Het zou wel mooi zijn om een formulier te hebben waarin we alle titels per uitgever zouden kunnen raadplegen.

☼ Maak voor Boekenier een query die een tabel maakt en vult met Uitgevers + Boektitels + Schrijver + ISBN-code. Haal de Uitgever uit tblUitgevers en niet tblISBN, want in de laatste staan de uitgevers als ID opgenomen en niet direct per naam. Noem de tabel maar tblTitelsPerUitgever. Bewaar hem maar even onder de naam die Access zelf kiest, Query1 of zoiets. We kunnen hem straks hergebruiken!

Je query moet er ongeveer zo uitzien:

☼ Maak een query die de tabel leegt (**qryTitelsPerUitgeverLeeg**)

☼ Maak een query die de tabel vult met informatie voor de uitgever Florada. Gebruik hiervoor die eerdere tabelmaakquery weer. Verander daarna de naam van de query in **qryTitelsPerUitgeverVul**. (Via selecteren van Query1 in het databasevenster, F2 en veranderen)

☼ Voeg een nieuw formulier toe **frmTitelsPerUitgever** en plaats daarop een knop(cmdTitelsPerUitgever), die de tabel **tblTitelsPerUitgever** leegt en weer vult via **qryTitelsPerUitgeverVul**. Zorg ervoor dat er geen waarschuwingen getoond worden.

☼ Voeg aan de procedure aan het eind de volgende opdracht toe:
DoCmd.OpenTable "tblTitelsPerUitgever"

Deze laatste opdracht laat de tabel zien.

☼ Probeer

Nou willen we ook wel eens een lijstje van een andere uitgever zien. Ga straks die query maar eens aanpassen met een andere uitgever als criterium. Je zult zien dat als je hem uitvoert vanuit de knop op het formulier, het tabelvenster niet ververst wordt. Daar moeten we dus zo iets aan gaan doen.

☼ Probeer dit maar eens.

De oplossing is om de tabel te sluiten voordat hij opnieuw gevuld en getoond wordt.

De opdracht hiervoor is: **DoCmd.Close acTable, "tblTitelsPerUitgever"**

Het is jammer m.i. dat Access het openen en sluiten van tabellen, formulieren en rapporteren met een andere opdrachtstructuur doet, vergelijk maar:

Sluiten:	**DoCmd.Close**	**acTable, "tblTitelsPerUitgever"**
Openen:	**DoCmd.OpenTable**	**"tblTitelsPerUitgever"**

Bij het sluiten wordt een algemene methode "Close" gebruikt en daarna moet je met zgn. parameters aangeven wat je wilt sluiten. Bij het openen heb je verschillende methoden: **OpenTable/OpenForm/OpenReport** etc. en hoef je alleen maar aan te geven om welk specifiek rapport/tabel/formulier etc. het gaat. We zullen er wel mee moeten leven.

☼ Voeg de sluitopdracht toe aan het begin van de procedure en voer nogmaals uit.

☼ Verander de uitgever nog eens in de query en voer weer uit.

Telkens die query handmatig aanpassen is natuurlijk geen automatisering. We gaan in de volgende ronde die opdrachtknop vervangen door een lijstknop, waarachter alle uitgevers gehangen worden. Na keuze van een uitgever willen we dan dat die query automatisch aangepast wordt zodat we de gewenste gegevens krijgen.

Vijfde ronde

Sessie 5 - 1

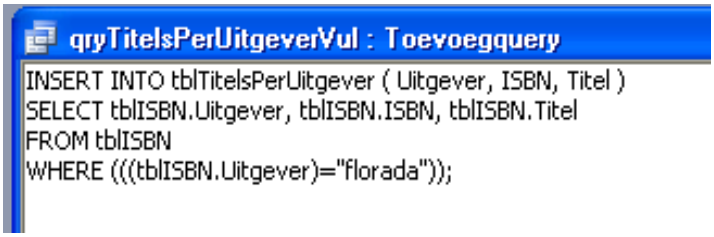
Queries in code uitvoeren.

☼ We willen er naartoe dat we een uitgever kunnen kiezen en dat de query dan automatisch aangepast wordt.

☼ Open nu de **qryTitelsPerUitgevervul** in ontwerpmodus.

☼ Kies uit het menu **Beeld / SQL**

Je krijgt de onderliggende sql-code te zien (wellicht met een andere uitgevernaam):



Op de achtergrond worden alle queries in dit SQL-taaltje gegoten en het is wel handig voor een access-programmeur om hier een beetje vanaf te weten, maar ook met een minimale kennis kun je er handig gebruik van maken. We gaan die tekst eens kopiëren en plakken in onze code. Je zult een hoop rode fout waarschuwingen krijgen, maar die kun je telkens wel negeren.

☼ Terug naar de code van je opdrachtknop. Type eerst : sql01 = " op de plek hieronder aangegeven en kopieer all sql-tekst en plak die hierachter. Je krijgt veel rood en foutmeldingen.

```
Private Sub cmdTitelsPerUitgever_Click()
DoCmd.SetWarnings False
DoCmd.Close acTable, "tblTitelsPerUitgever"
DoCmd.OpenQuery "qryTitelsPerUitgeverLeeg", , acReadOnly
sql01 = "   <--------
sqlstring = sql01 & sql02 & sql03
DoCmd.RunSQL sqlstring
DoCmd.OpenTable "tblTitelsPerUitgever"
End Sub
```

☼ Verwijder de Enters en de aanhalingstekens zodat je achter sql01= " het volgende krijgt:

```
sql01 = "INSERT INTO tblTitelsPerUitgever ( Uitgever, ISBN, Titel )SELECT tblISBN.Uitgever, tblISBN.ISBN, tblISBN.Titel FROM tblISBN WHERE (((tblISBN.Uitgever)="
```

☼ Zorg nu onder sql01 voor:
sql02 = "Florada" (of welke naam ook maar in je sql-code staat)

☼ En daar weer onder de rest van je oorspronkelijke tekst toegekend aan sql03.
sql03 = "));"

Omdat de aanhalingstekens voor de naam van de uitgever, bijv. "Florada", niet meer doen dan aangeven dat het om een tekst (een string) gaat, zou je als je straks de boel aan elkaar gaat plakken het volgende krijgen: ….. **WHERE (((tblISBN.Uitgever)=Florada));** **Florada** is nu niet meer omringd door aanhalingstekens en de code zal niet goed werken. Met dubbel aanhalingstekens los je dit ook niet op: sql02 = ""Florada"" gaat ook niet helpen. De enige manier om dit op te lossen is gebruik te maken van enkele aanhalingstekens en ze in de variabelen voor en na de tekstvariabele te plaatsen.

☼ Voeg een enkele aanhalingsteken toe aan het eind van de tekst in sql01:
WHERE (((tblISBN.Uitgever)=" → WHERE (((tblISBN.Uitgever)='"

☼ Voeg ook een enkele aanhalingsteken toe aan het begin van de tekst in sql03:
"));" → "'));"

☼ Plak alles hieronder weer aan elkaar: sqlstring = sql01 & sql02 & sql03 en bovenaan voor de netheid even alle variabelen dimensioneren/declareren.

☼ En ten slotte moet je de openquery-regel vervangen door *DoCmd.RunSql sqlstring*

Je krijgt:

```
Private Sub cmdTitelsPerUitgever_Click()
Dim sqlstring As String
Dim sql01 As String
Dim sql02 As String
Dim sql03 As String
DoCmd.SetWarnings False
DoCmd.Close acTable, "tblTitelsPerUitgever"
DoCmd.OpenQuery "qryTitelsPerUitgeverLeeg", , acReadOnly
sql01 = "INSERT INTO tblTitelsPerUitgever ( Uitgever, ISBN, Titel )SELECT tblISBN.Uitgever, tblISBN.ISBN, tblISBN.Titel FROM tblISBN WHERE (((tblISBN.Uitgever)='"
sql02 = "Florada"
sql03 = "'));"
sqlstring = sql01 & sql02 & sql03
DoCmd.RunSQL sqlstring
DoCmd.OpenTable "tblTitelsPerUitgever"
End Sub
```

Ok wat gebeurt hier? De RunSql methode laat een query-uitvoeren die in SQL gecodeerd is. Nou heb ik die code in stukjes verdeeld en aan variabelen toegekend en wel zo dat het stukje dat we flexibel willen hebben (hier de naam van de uitgever) in een aparte variabele komt te zitten. Door die variabele (hier sql02) met een andere uitgever te vullen, kunnen we dus de query wijzigen.

Nadat we de query in stukjes hebben geknipt, plakken we hem weer aan elkaar. Tekst (strings) plak je aan elkaar met "&" en niet "+". Uiteindelijk zit dezelfde sqltekst weer in sqlstring nadat de drie delen aan elkaar zijn gelijmd. Die kunnen we nu uit laten voeren.

☼ Kijk even of alles nog werkt voordat we verder gaan. Type bijv. eens een andere uitgever in.

Nu er nog voor zorgen dat we die naam niet steeds in de code aan hoeven te passen, maar bijv. uit een lijstje kunnen halen.

☼ Verwijder uit je formulier de opdrachtknop "**cmdTitelsPerUitgever**"

☼ Plaats een keuzelijst (**kzlUitgever**) die je uitgevers laat kiezen (baseer op **tblUitgevers**), geef de label "Titels per uitgever" mee.

☼ Zorg voor het klik-event en ga dan naar je vbacode voor dit formulier.

☼ Kopieer de code van **cmdTitelsPerUitgever_Click** naar **kzlUitgever_Click**. Verwijder daarna de volledige **cmdTitelsPerUitgever_Click** procedure.

☼ Probeer de keuzelijst even om te zien of alles nog werkt. Er wordt nog geen keuze gemaakt!

☼ Pas je code nu als volgt aan en probeer telkens na het kiezen van een andere uitgever.

Private Sub kzlUitgever_Click()
sql02 = Me.kzlUitgever.column(1)

Verwijder: sql02 = "florada" (of wat daar ook maar staat)

Een voordeel van het opslaan van geselecteerde gegevens in een tijdelijke tabel is dat je er daarna nog van alles mee kunt doen, bijv. een rapport ervan maken, gebruiken als basis voor een mailing etc.

Sessie 5 - 2

Oefening:

1.
Voeg een keuzelijst toe onder **kzlUitgever** met de naam **kzlSchrijvers** (label: *Titels per schrijver*) dat je kunt gebruiken om per schrijver een tabellijstje te krijgen van zijn boeken + de prijs ervan.

Sessie 5 - 3

2.
In het hoofdstedenspel kun je nu de zaak flessen door zonder eerst op volgende te klikken de goede hoofdstad uit te kiezen nadat je een fout hebt gemaakt.. Je score vliegt dan nog een stukje onverdiend omhoog. Probeer maar eens. Bedenk een manier waarbij je een globale variabele gebruikt om te onthouden of er al een keer gekozen is, zodat je maar één keer een hoofdstad kunt selecteren nadat je op Volgende hebt geklikt. Laat de gebruiker weten dat er niet gefraudeerd mag worden. Je zou eventueel gebruik kunnen maken van een zogenaamde boolean variabele waarin alleen True of False mag staan. Je definieert die als volgt (Ik noem hem maar "geweest|"):

Dim geweest as boolean

En daarna kun je zeggen: geweest = True of geweest = False

Sessie 5 - 4

Vergelijkingsoperatoren

☼ Open (zo nodig) Boekenier.mdb

Opdracht

☼ Maak een query (**qryBestellen**)die laat zien welke boeken aan bestellen toe zijn (**Voorraad <= [Besteldrempel]),** d.w.z. waar de voorraad gelijk of minder is dan de Besteldrempel. Zorg er ook voor dat je alle bestelgegevens er direct bij te zien krijgt.

We kunnen nu verschillende formulieren maken, bijv. één met alleen de telfoonnummers erbij voor telefonische bestelling, één met de adres, postcode n plaats gegevens voor bestelling per post of één voor bestelling per email.

☼ Maak een Formulier (**frmQryBestellen**) met alleen de e-mailadressen erbij, zoiets:

Hier is een lijstje van zgn. **Vergelijkingsoperatoren:**

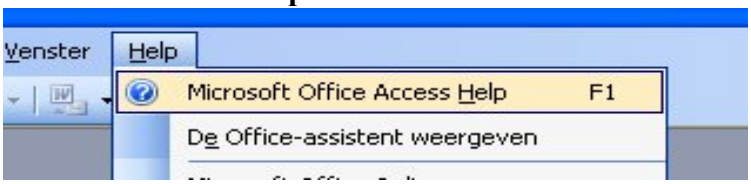
Operator
< (Kleiner dan)
<= (Kleiner of gelijk aan)
> (Groter dan)
>= (Groter of gelijk aan)
= (Gelijk aan)
<> (Niet gelijk aan)

HELP gebruiken

Je kunt dit soort overzichten ook zelf wel vinden via je de zoekfunctie in Access. Zoek bijv. op **Operator**

Probeer maar te volgen.

☼ Kies **Help/Microsoft Office Access Help**



☼ Type in operator en klik op de groene pijl.

☼ Kies **Vergelijkingsoperatoren** (kan even zoeken zijn)

☼ Open de keuze *Overzicht van operators voor zoekcriteria.*

Zesde ronde

Sessie 6 - 1

Zoeken via dynamische queries

☼ Kijk nog eens naar je formulier frmUitgever en de bijbehorende code. Dit formulier moet het adres van een uitgever uit de database vissen en tonen. Om het juiste adres te vinden laten we iedere record als volgt controleren:

```
info = rec.fields("Uitgever")
If  info = keuze then
```

Als het om heel veel data gaat is deze manier van zoeken erg langzaam. VBA kent nog wel meer zoekmethoden, maar het SQL-object van Access werkt het snelst en nu we queries dynamisch aan kunnen passen, kunnen we daar mooi gebruik van maken.

☼ Maak een kopie van de procedure **Private Sub toon_gegevens(keuze)** zodat we de oude situatie ook nog even gemakkelijk terug kunnen halen, mocht dat nodig zijn. Verander de naam van de oude in **Private Sub toon_gegevens_oud(keuze)**

☼ Maak een query die de gewenste adresgegevens uit de tabel **tblUitgevers** haalt met als criterium bijv. de uitgever "Uithoek". Kopieer en plak de gegevens in de aan te passen procedure **Private Sub toon_gegevens(keuze)** zodat je het volgende krijgt (let op: de tekst achter sql01 = moet achter elkaar aan op één regel). Zorg ook voor die extra enkele aanhalingsteken in sql01 en sql03. Let er verder op dat ik overal **rec** door **qrec** vervangen heb. Dit hoeft niet, is wat meer werk, maar wellicht wel de moeite waard om later direct te kunnen zien dat het hier om een geopende query gaat. Pas de procedure ook verder aan, d.w.z. haal weg wat niet langer nodig is. Merk op dat ik de gegevens nu direct in de labels plaats (en niet meer via de tussenvariabele Info). Niet nodig maar korter.

```
Private Sub Uitgever_adres(keuze)
Dim db As Database
Dim qrec As Recordset
Set db = CurrentDb()
sql01 = "SELECT tblUitgevers.Uitgever, tblUitgevers.Adres, tblUitgevers.Postcode,
        tblUitgevers.Plaats, tblUitgevers.Telefoon, tblUitgevers.Email FROM tblUitgevers
        WHERE (((tblUitgevers.Uitgever)='"
sql02 = keuze
sql03 = "'));"
sqlstring = sql01 & sql02 & sql03
Set qrec = db.OpenRecordset(sqlstring)
lblNaam.Caption = qrec.Fields("Uitgever")
lblAdres.Caption = qrec.Fields("Adres")
lblPc.Caption = qrec.Fields("Postcode")
lblPlaats.Caption = qrec.Fields("Plaats")
qrec.Close
End Sub
```

☼ Probeer

TWIPS

Er ontstond behoefte aan het hebben van een schermmaat die niet afhankelijk was van monitor resoluties etc., maar dezelfde afmetingen zou hebben op iedere monitor. Op een 15" monitor dus even groot als op een 24" monitor, ongeacht de resolutie van die schermen. En dat werd de TWIP (Twentieth of an Inch Point) of eenvoudigweg: een twip is 1/567 ste van een centimeter. Lekker makkelijk dus!

Je kunt die TWIPS van een actief venster met de volgende opdracht aanpassen.

DoCmd.MoveSize ***naarrechts***, ***naaromlaag***, ***breedte***, ***hoogte***

Met die eerste twee parameters, kun je hem dus verschuiven en met de laatste twee aanpassen. Je moet bij dit soort commando's de plek van die eerdere parameters altijd vrijhouden. Om de tabel bijv. 5000 twips breed en 5000 twips hoog te maken gebruik je:

DoCmd.MoveSize , , ***5000***, ***5000***

☼ Plaats deze opdracht aan het eind van je *toon_gegevens-procedure* en probeer het weer.

Opdracht:

☼ Zorg ook voor een vaste grootte van de tabellen **tblTitelsPerSchrijver** en **tblTitelsPerUitgever** op het moment dat ze getoond worden.

Oplossing:

Voeg de DoCmd-code toe als volgt:

```
Private Sub kzlUitgever_Click()
sql02 = kzlUitgever.Column(1)
DoCmd.SetWarnings False
DoCmd.Close acTable, "tblTitelsPerUitgever"
DoCmd.OpenQuery "qryTitelsPerUitgeverLeeg"
sql01 = "INSERT INTO tblTitelsPerUitgever ( Uitgever,
sql03 = "'));"
sqlstring = sql01 & sql02 & sql03
DoCmd.RunSQL sqlstring
DoCmd.OpenTable "tblTitelsPerUitgever"
DoCmd.MoveSize , , 7000, 5000
End Sub
```

Ook voor de kzlSchrijver-procedure. Bepaal zelf de grootte.

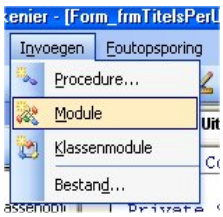
Sessie 6 - 2

Modules en gedeelde procedures

De procedures in **frmKeuzelijsten** lijken erg veel op elkaar. We gaan nu naar een manier kijken om delen van procedures die veel op elkaar lijken en vaak zelfs in verschillende formulieren voorkomen op één plek te centraliseren. Omdat de procedures zelden helemaal identiek zijn, sturen we dan de variabele informatie mee.

Module toevoegen

☼ Kies in de VBA-editor voor **Invoegen** en **Module**



☼ Zodra hij gemaakt is selecteer je de module in het projectverkennervenster, klik dan op **Beeld** en **Venstereigenschappen**. Daar kun je zelf een naam geven aan de module. Noem hem maar **SQLverwerken**

☼ Sluit het eigenschappenvenster

☼ Voeg een nieuwe procedure toe in deze module:

```
Sub SQLuitvoeren()

End Sub
```

☼ Kopieer nu de code uit bijv. **kzlSchrijver_Click** uit naar deze procedure en verander de code dan als volgt:

```
Sub SQLuitvoeren(tabelnaam, querynaam, sqlstring)
DoCmd.SetWarnings False
DoCmd.Close acTable, tabelnaam
DoCmd.OpenQuery querynaam
DoCmd.RunSQL sqlstring
DoCmd.OpenTable tabelnaam
DoCmd.MoveSize , , 7000, 5000   '(of wat je hier maar hebt aan twips)
End Sub
```

☼ Pas vervolgens de code in **kzlSchrijver_Click** als volgt aan (tekst van sql01 op één regel!):

```
Private Sub kzlSchrijver_Click()
Dim sqlstring As String
Dim tabelnaam As String
Dim querynaam As String
sql02 = kzlSchrijver.Column(0)
tabelnaam = "tblTitelsPerSchrijver"
querynaam = "qryTitelsPerSchrijverLeeg"
sql01 = "INSERT INTO tblTitelsPerSchrijver ( Schrijver, ISBN, Titel, Categorie ) SELECT
        tblISBN.Schrijver, tblISBN.ISBN, tblISBN.Titel, tblISBN.Categorie FROM tblISBN
        WHERE (((tblISBN.Schrijver)='"
sql03 = "'));"
sqlstring = sql01 & sql02 & sql03
SQLverwerken tabelnaam, querynaam, sqlstring
End Sub
```

Wat doen we hier? We kennen alle variabele gegevens toe aan variabelen. De tabelnaam aan de variabele **Tabelnaam**, de leegquery aan **Querynaam** en de sqlstring laten we staan omdat die de sqlcode al aan de variabele **Sqlstring** toekent. Voor de netheid declareren we deze variabelen ook nog even (in de video wordt dat niet gedaan!)

Met de laatste opdracht sturen we de info door naar de procedure in onze module die al het zware werk gaat doen. De variaben tabelnaam, querynaam en sqlstring worden in dezelfde volgorde doorgegeven aan de variabelen in SQLuitvoeren. Let daar even op, want de naam is niet belangrijk voor het doorgeven, wel de positie. SQLuitvoeren voert dan alle acties uit.

☼ Pas de code in kzlUitgever_click nu op dezelfde manier aan.

```
Private Sub kzlUitgever_Click()
sql02 = kzlUitgever.Column(1)
tabelnaam = "tblTitelsPerUitgever"
querynaam = "qryTitelsPerUitgeverLeeg"
sql01 = "INSERT INTO tblTitelsPerUitgever ( Uitgever, Titel, Schrijver, ISBN )SELECT
        tblUitgevers.Uitgever, tblISBN.Titel, tblISBN.Schrijver, tblISBN.ISBN FROM
        tblUitgevers INNER JOIN tblISBN ON tblUitgevers.Id = tblISBN.Uitgever WHERE
        (((tblUitgevers.Uitgever)='"
sql03 = "'));"
sqlstring = sql01 & sql02 & sql03
sqluitvoeren tabelnaam, querynaam, sqlstring
End Sub
```

☼ Probeer.

Sessie 6 - 3

Oefening

☼ Open **OefenDB** (of hoe je hem ook maar genoemd hebt)

☼ Voeg een module toe en noem die maar **Meldingen**

☼ Plaats hierin de volgende code:

```
Sub Melding(boodschap)
MsgBox (boodschap)
End Sub
```

☼ Zorg er nu voor dat overal waar je een Msgbox gebruikt (in frmHoofdsteden zowel als frmTest) die vervangen wordt door code die deze routine aanroept. (Opmerking: Je code wordt er niet korter door, maar het gaat hier even om toepassing van de techniek)

Sessie 6 - 4

Functies

Functies zijn procedures die een waarde retourneren naar de aanroepende procedure. Bij gewone Sub's kun je variabelen met inhoud doorsturen naar ander procedures. Bij functies kan dat ook maar krijg je er ook iets voor terug.

☼ Voeg (in je **OefenDB**) nog een module toe en noem die maar **Functies**

☼ Type hierin de volgende code:

```
Function willekeurig(hoogste)
willekeurig = Int((hoogste * Rnd) + 1)
```

```
End Function
```

☼ Ga nu naar je frmTest code en de module cmdNieuwe_Click

☼ Verander de code in:

```
Private Sub cmdNieuwe_Click()
Randomize
getal1 = willekeurig(100)
getal2 = willekeurig(100)
txtGetal1.SetFocus
txtGetal1.Text = getal1
txtGetal2.SetFocus
txtGetal2.Text = getal2
End Sub
```

Uitleg: Getal1 = willekeurig(100) betekent: roep de functie willekeurig aan en stuur 100 mee. Die 100 komt in de variabele **hoogste** terecht van die functie. Op basis van het getal in **hoogste** wordt een willekeurig getal gegenereerd en dat wordt in de variabele **willekeurig** geplaatst. **Willekeurig** is hetzelfde als **willekeurig(100)**. Die **willekeurig(100)** functioneert dus aan de ene kant als een oproep naar een functie en tegelijkertijd als een variabele. In **willekeurig(100)** komt dus een waarde te zitten en die waarde wordt weer doorgegeven aan **getal1** en daarna een keer aan **getal2**.

☼ Probeer en zet even ergens een breekpunt zodat je de code stapje voor stapje doorneemt en begrijpt wat er gebeurt.

Sessie 6 – 5

Oefening:

☼ Pas het nu ook toe voor **frmHoofdsteden**. Je moet hier oppassen en je goed realiseren wat er gebeurt. Je stuurt als **Hoogste** het aantalrecords mee (bijv. 40). De functie retourneert dan een getal van 1 t/m 40. Maar de records beginnen te tellen bij 0 en houden dan op bij 39. Hier moet je rekening mee houden in je code. Succes!

Zevende ronde

Sessie 7 - 1

Interface bouwen

Om gebruikers toegang te verlenen tot de diverse formulieren moet je een zgn. interface bouwen. Zo'n interface bestaat uit een formulier of formulieren van waaruit de diverse onderdelen te benaderen zijn. Voor Boekenier zou je het volgende op kunnen zetten:

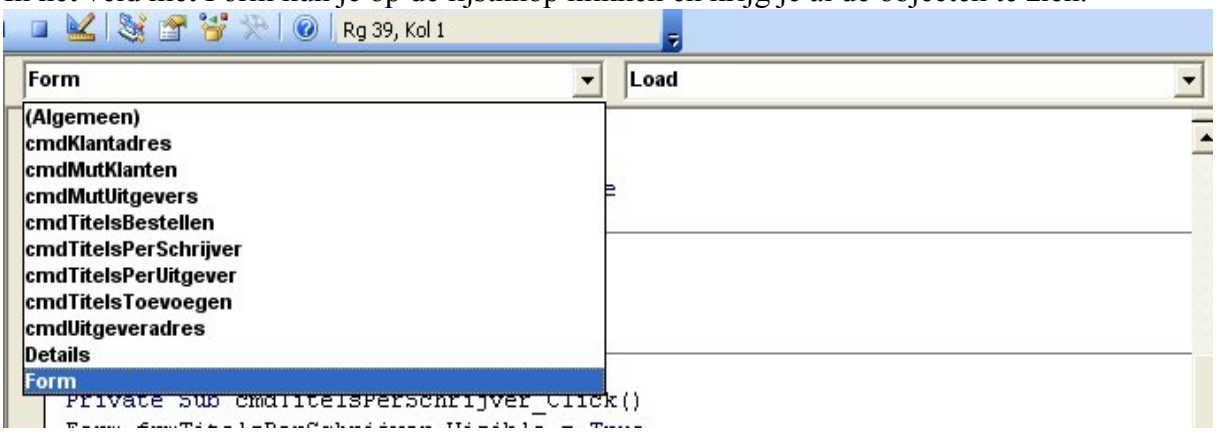
- ☼ Start een nieuw formulier, nergens op gebaseerd.
- ☼ Plaats één knop met ongeveer de gewenste grootte.
- ☼ Selecteer de knop. Ctrl/C en dan 7 keer Ctrl/V
- ☼ Je moet nu 8 knoppen hebben.
- ☼ Plaats ze ongeveer zoals in het voorbeeld.
- ☼ Voeg nu per knop de teksten (bijschriften) toe en geef ze de volgende namen:

cmdKlantadres
cmdMutKlanten
cmdMutUitgevers
cmdTitelsBestellen
cmdTitelsPerSchrijver
cmdTitelsPerUitgever
cmdTitelsToevoegen
cmdUitgeveradres

☼ Sla het Formulier op en noem het **frmStart**

☼ Open nu de Editor en het formulier **frmStart**

In het veld met Form kun je op de lijstknop klikken en krijg je al de objecten te zien.



☼ Klik ze allemaal aan met het Click-event (rechts) en vul ze daarna als volgt: (Je zou één kunnen doen en die daarna kopiëren, plakken en aanpassen). (*Het formulier frmQryBestellen is nog niet zichtbaar, doen we \zo iets aan*)

```
Private Sub cmdMutKlanten_Click()
MsgBox ("Nog niet toegevoegd")
End Sub

Private Sub cmdMutUitgevers_Click()
Form_frmMutatieUitgevers.Visible = True
End Sub
```

Het hier genoemde formulier is nog niet bekend bij VBA, maar vul de verwijzing toch als vast maar in:

```
Private Sub cmdTitelsBestellen_Click()
Form_frmQryBestellen.Visible = True
End Sub
```

De volgende twee knoppen laten we hetzelfe formulier openen:

```
Private Sub cmdTitelsPerSchrijver_Click()
Form_frmKeuzelijsten.Visible = True
End Sub

Private Sub cmdTitelsPerUitgever_Click()
Form_frmKeuzelijsten.Visible = True
End Sub
```

```
Private Sub cmdTitelsToevoegen_Click()
Form_frmVoegtoe.Visible = True
End Sub

Private Sub cmdUitgeveradres_Click()
Form_frmUitgever.Visible = True
End Sub
```

Formulier bij VBA bekend maken

VBA moet formulier kennen om er iets mee te kunnen doen. Die bekendheid komt pas op het moment dat er een event aan zo'n formulier gekoppeld geweest is.

☼ Je kunt snel een gebeurtenis koppelen aan een formulier door in de ontwerpweergave met de rechtermuisknop op de knop links boven te klikken en dan te kiezen voor **Gebeurtenis opbouwen** etc.

☼ Zorg voor de gebeurtenis (Load!)

frmQryBestellen moet nu ook in het projectvenster zichtbaar zijn.

☼ Probeer de knoppen.

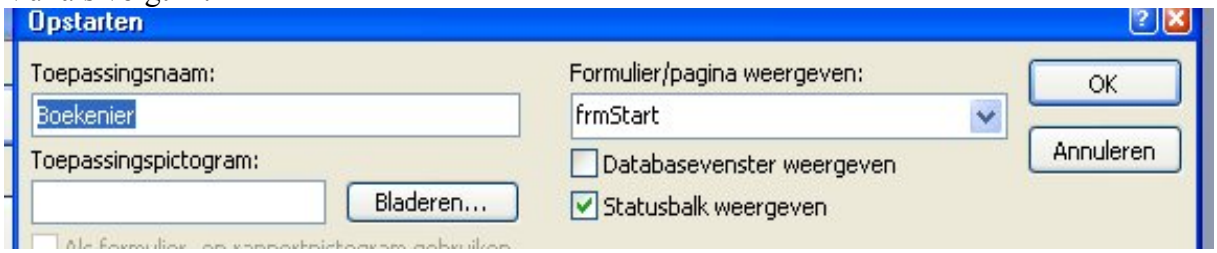
Sessie 7 - 2

Startformulier instellen

Om een formulier in te stellen als start formulier volg je deze stappen:

☼ In Access zelf: **Extra / Opstarten**

☼ Vul als volgt in:



Let op het Databasevenster niet weergeven is niet echt handig voor de ontwikkelaar, maar je wilt dit misschien wel doen als anderen met de applicatie moeten gaan werken.

☼ Sluit de Applicatie en start hem weer op.

☼ Voeg het Databasevenster weer toe.

Je moet voorzichtig zijn met het niet tonen van de menu's (dat kan ook in dat Opstartenvenster) want voor dat je het weet kun je niet meer bij je eigen programmacode.

Positie, weergave en grootte van formulieren bepalen.

Het handigst is waarschijnlijk de formuliereigenschappen: AutoCentreren en AutoWijzigen op "Ja" te zetten. Je krijgt dan formulieren die gecentreerd getoond worden en waarvan de grootte wordt aangepast aan de inhoud. Je doet dat bij de formuliereigneschappen, we komen daar zo op terug.

Mocht je de positie per formulier zelf willen bepalen dan kun je dat ook in code doen. Gebruik dan de DoCmd.MoveSize opdracht, bijv. voor het frmStart-formulier. Gebruik hiervoor het Form – Load event:

☼ Zorg voor de volgende code voor het frmStarten formulier en probeer:

```
Private Sub Form_Load()
DoCmd.MoveSize 300,300
End Sub
```

Echt nodig is dit waarschijnlijk niet, zie dus maar wat je doet.

Verder willen we ook wel graag van allerlei andere aspecten van de formulieren verlost worden. Je maakt het jezelf het gemakkelijkst door de volgende eigenschappen van de formulieren als volgt in te stellen:

Recordkiezers	Nee
Navigatieknoppen	Nee
Recordbegrenzingslijnen	Nee
AutoWijzigen formaat	Ja
AutoCentreren	Ja

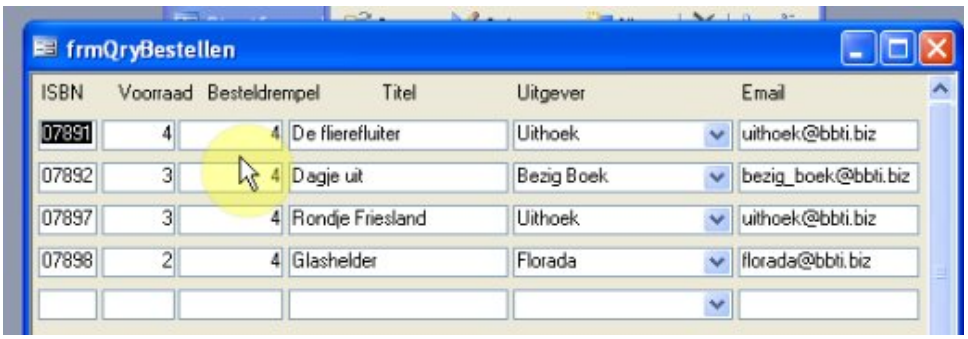
☼ Doe maar voor alle formulieren!

☼ Opdracht:

Probeer zelf eens uit te vinden waar deze eigenschappen precies naar verwijzen, bijvoorbeeld door telkens één van de eerste drie op “ja” te zetten.

Sessie 7 - 3

Formulier records lezen (gekloonde recordset)



Zodra je een formulier gekoppeld hebt aan een set gegevens (een tabel/query-resulktaten etc) kun je met de navigatieknoppen in dat formulier door de records bladeren, maar hoe kun je in VBA deze records aanspreken. Een veelgebruikte methode is om van die records een kloon te maken in de vorm van een recordset. Die recordset kun je dan op de gebruikelijke manier mee omgaan.

☼ Set een opdrachtknop op je formulier met de naam **cmdMail** en opschrift bijv. **Stuur mail**

☼ Koppel er een klikgebeurtenis aan.

☼ Voer de volgende code in in je procedure:

```
Private Sub cmdMail_Click()
Dim rs As Recordset
Set rs = Me.RecordsetClone
aantal = rs.RecordCount
rs.MoveFirst
For x = 1 To aantal
Mail = rs("email")
Uitgever = rs("uitgever")
boektitel = rs("titel")
isbnnr = rs("ISBN")
rs.MoveNext
Next
rs.Close
End Sub
```

Merk op dat ik hier gebruik maak van **rs** in plaats van **rec** ! Daar zit niets achter, ik had ook **rec** kunnen gebruiken. Het is niet meer dan een verkorte aanduiding voor een recordset. Ik heb **rs** hier gebruikt omdat ik deze en de volgende emailcode van het Internet heb geplukt. Iets wat je ook zult moeten doen als je serieus aan het programmeren gaat en dan kom je dus allerlei variaties tegen, waar je niet van moet schrikken.

☼ Plaats een stop in Mail = rs("email") en voer je code dan uit. Je zult zien dat alle record gegevens één voor één getoond worden.

Sessie 7 - 4

Vanuit Access e-mail versturen

Je kunt via VBA gebruik maken van een zgn Outlook-object om berichten de deur uit te doen naar e-mailadressen in je database, compleet met onderwerp en bijlage (indien gewenst).

☼ Pas je code als volgt aan (*merk op de een tekst voorafgegaan door een enkele aanhalingsteken, niet meer is dan een opmerking*):

```
Private Sub cmdMail_Click()
onderwerp = "Bestelling"
aanhef = "Geachte heer, mevrouw, "
bijlage = ""
Dim rs As Recordset
Set rs = Me.RecordsetClone
aantal = rs.RecordCount
rs.MoveFirst
For x = 1 To aantal
Set EmailApp = CreateObject("Outlook.Application")
Set NameSpace = EmailApp.GetNamespace("MAPI")
Set EmailSend = EmailApp.CreateItem(0)
Mail = rs("email")
Uitgever = rs("uitgever")
boektitel = rs("titel")
isbnnr = rs("ISBN")
bericht = "T.a.v. " & Uitgever & vbCrLf & aanhef & vbCrLf & vbCrLf & "Graag ontvangen
wij 3 exemplaren van:" & vbCrLf & boektitel & vbCrLf & "ISBN: " & isbnnr
EmailSend.To = Mail ' Put email address here
EmailSend.Subject = onderwerp
EmailSend.Body = bericht
If bijlage = "" Then
Else
EmailSend.Attachments.Add bijlage ' Change this to match your path
End If
EmailSend.display ' verander in Emailsend.send om de berichten niet eerst te zien
Set EmailApp = Nothing
Set NameSpace = Nothing
Set EmailSend = Nothing
rs.MoveNext
Next
rs.Close
End Sub
```

In de volgende code wordt alles ingericht voor verzending:

EmailSend.To = Mail ' Put email address here
EmailSend.Subject = onderwerp
EmailSend.Body = bericht

Aan **mail** hadden we het e-mailadres van het record dat aan de beurt was toegekend, in **onderwerp** vermeld je het onderwerp van het bericht en in de variabele **bericht** zelf komt de tekstboodschap te staan die je wilt vesturen. Ik kom daar zo nog weer op terug.

Merk op dat dit allemaal eigenschappen zijn, het toekennen van een bijlage gaat met een methode:

EmailSend.Attachments.Add bijlage

In bijlage staat de naam en het pad van je bijlage, bijv: C://Mijn documenten/nieuwsbrief.pdf
EmailSend.display

Deze methode stuurt het bericht naar Outlook, waar je het kunt bekijken en zo nodig wijzigen voordat je het verstuurt. Je kunt ook dit gebruiken:

EmailSend.send

Doe je dat dan gaat de post gelijk de deur uit en heb je niet meer de gelegenheid om de berichten eerst te controleren. Doe dat dus pas als je weet dat alles in orde is.

Samenstelling van de tekst body

De zgn body van de tekst, het bericht zelf dus, moet je nauwkeurig samen stellen. Dat van ons ziet er zo uit:

bericht = "T.a.v. " & Uitgever & vbCrLf & aanhef & vbCrLf & vbCrLf & "Graag ontvangen wij 3 exemplaren van:" & vbCrLf & boektitel & vbCrLf & "ISBN: " & isbnnr

De tekst wordt toegekend aan de variabele bericht en bestaat uit stukjes tekst (strings) omgeven door aanhalingstekens, variabelen met relevante onderdelen van je bericht en vba constanten. VBA-constanten werken als variabelen maar hebben een voorgeprogrammeerde betekenis. vbCrLF zorgt ervoor dat er een ControlLineFeed uitgevoerd wordt of in een mensen taal, een Enter toegepast wordt waarmee je naar een nieuwe regel gaat.
Al die onderdelen worden met het &-teken aan elkaar geplakt.

Je zou de emailadressen in boekenier kunnen vervangen door eigen adressen of/en die van vrienden die het niet erg vinden eens een test-mailtje te ontvangen. Op het moment van schrijven kun je de gegeven e-mailadressen ook wel gebruiken. Zorg er dan wel voor dat de extensie .biz is en niet .nl (zoals in eerdere versies), bijv. florada@bbti.biz

☼ Probeer

Succes gewenst!